No. 2016-1402

UNITED STATES COURT OF APPEALS FOR THE FEDERAL CIRCUIT

IN RE APPLE INC.,

Appellant.

Appeal from the United States Patent and Trademark Office, Patent Trial and Appeal Board in Reexamination Control No. 90/012,332

BRIEF FOR APPELLANT APPLE INC.

RACHEL KREVANS
NATHAN B. SABRI
MORRISON & FOERSTER LLP
425 Market Street
San Francisco, CA 94105
(415) 268-7000

WILLIAM F. LEE
DOMINIC E. MASSA
LAUREN B. FLETCHER
SYDENHAM B. ALEXANDER, III
WILMER CUTLER PICKERING
HALE AND DORR LLP
60 State Street
Boston, MA 02109
(617) 526-6000

May 3, 2016

Attorneys for Appellant Apple Inc.

CERTIFICATE OF INTEREST

Counsel for Appellant Apple Inc. certifies the following:

1. The full name of every party or *amicus* represented by us is:

Apple Inc.

2. The names of the real party in interest represented by us is:

Not applicable.

3. All parent corporations and any publicly held companies that own 10 percent or more of the stock of the party or amicus curiae represented by me are:

None.

4. The names of all law firms and the partners or associates that appeared for the party or amicus now represented by me in the trial court or agency or are expected to appear in this court are:

WILMER CUTLER PICKERING HALE AND DORR LLP: Sydenham B. Alexander, III, Lauren B. Fletcher, Sarah R. Frazier, William F. Lee, Dominic E. Massa

MORRISON & FOERSTER LLP: Brian B. Ho, Richard S.J. Hung, Rachel Krevans, Nathan B. Sabri, Peter J. Yim

Dated: May 3, 2016

/s/ William F. Lee

WILLIAM F. LEE
WILMER CUTLER PICKERING
HALE AND DORR LLP
60 State Street
Boston, MA 02109
(617) 526-6000

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

CASES

Page(s)

DOCKETED CASES

STATUTES AND REGULATIONS

STATEMENT OF RELATED CASES

This appeal concerns reexamination proceedings for U.S. Patent No. 7,844,915 ("the '915 patent"). No appeal in this case was previously before this Court or any other court.

Certain remand proceedings in *Apple Inc. v. Samsung Electronics Co.*, No. 11-cv-1846 (N.D. Cal), could potentially be affected by the Court's decision in this appeal. In that case, a jury found (among other things) that claim 8 of Apple's '915 patent was infringed and not invalid, and awarded damages. This Court affirmed that judgment in its entirety with respect to the '915 patent. *See Apple Inc. v. Samsung Elecs. Co.*, 786 F.3d 983, 1003-1005 (Fed. Cir. 2015). That judgment is final and has been paid by Samsung; it will not be affected by this appeal. However, a retrial regarding additional damages issues, which could potentially include the '915 patent, has been postponed while the Supreme Court considers a question unrelated to the '915 patent in *Samsung Electronics Co. v. Apple Inc.*, No. 15-777 (certiorari granted to consider damages for design patent infringement).

The Supreme Court's decision in *Cuozzo Speed Technologies, LLC v. Lee*, No. 15-446, could affect this appeal, if the Supreme Court alters the "broadest reasonable interpretation" standard used for claim construction of an issued patent.

INTRODUCTION

The inventors of the ’915 patent created an intuitive way to control computers using touchscreens and our fingertips. Millions of people use the invention every day. It enables devices such as Apple’s iPhone and iPad to scroll, zoom, and rotate images—all in response to elegant touch commands that anyone can master. Although touchscreens were known in the prior art, no one had come up with a practical way for a touchscreen device to interpret and then respond to different types of touches. The inventors solved that problem by creating a single, simple rule for implementing a touchscreen device: interpret one finger touching the screen as a “scroll,” which can be used to slide content across the screen; interpret “two or more” fingers touching the screen as a “gesture,” which can be used to zoom in, zoom out, or rotate an image. Every claim in the ’915 patent uses this same rule to distinguish between single-touch scrolls and multiple-touch gestures.

The Patent Trial and Appeal Board (“the Board”) erroneously concluded that the invention was unpatentable because the Board misinterpreted the word “or.” The claims state that a touch with one finger is a scroll, and a touch with “two ***or*** more” fingers is a gesture. Based upon the word “or,” the Board construed the claims as not limited to a single rule for distinguishing between scrolls and gestures, but instead as extending to ***any*** rule that distinguishes between (1) a

single input point that is interpreted as a scroll operation and (2) two input points ***or*** more than two input points that are interpreted as a gesture operation. Under the Board's construction, the claims would be met by an algorithm that distinguishes between a single input point and two input points (but does not distinguish between a single input point and three input points) ***or*** by an algorithm that distinguishes between a single input point and three input points (but does not distinguish between a single input point and two input points). The Board's construction is not the "broadest reasonable interpretation" because it is inconsistent with the claim language and specification, both of which consistently describe a single rule for distinguishing between one-touch scrolls and multiple-touch gestures. It is also contrary to how a person of ordinary skill in the art would understand the claimed invention, as three experts testified but the Board disregarded.

The Board's construction divorced the claims from Apple's invention, and expanded the claims so much that the Board found them unpatentable in light of prior art that does not distinguish between a single-touch scroll and a three-touch gesture, but treats both as scrolls. The Board acknowledged that its claim construction was the basis for its decision and identified the construction as the "principal issue" in the reexamination. Appx6. Because the Board's decision is based upon an erroneous construction of the "scroll or gesture" limitation found in all claims, it should be reversed.

The Board also wrongly rejected three dependent claims based upon an erroneous construction of the term "rubberbanding." The inventors created a method of scrolling where "the content slides back" to its starting point when a user scrolls past a limit; they called this method "rubberbanding." The Board set aside the inventors' definition of "rubberbanding" and found that the claimed method was disclosed by prior art that did just the opposite—jumping ahead to new content when the user scrolls past a limit, rather than sliding back. Because that decision is also based upon an erroneous claim construction, it too should be reversed.

JURISDICTIONAL STATEMENT

The Board had jurisdiction pursuant to 35 U.S.C. §§ 134, 306. The Board issued its decision in the reexamination proceeding on December 9, 2014. Appx1-13. On September 24, 2015, the Board granted Apple's request for rehearing but declined to modify its decision. Appx14-21. Apple timely noticed its appeal on November 23, 2015. Appx935-936. This Court has jurisdiction pursuant to 28 U.S.C. § 1295(a)(4)(A).

STATEMENT OF ISSUES ON APPEAL

1. Whether the Board erred in determining that claims 1-21 of the '915 patent are unpatentable based upon its erroneous interpretation of the claim limitation for "determining whether the event object invokes a scroll or gesture

operation by ***distinguishing between a single input point*** applied to the touch-sensitive display that is interpreted as the scroll operation ***and two or more input points*** applied to the touch-sensitive display that are interpreted as the gesture operation."

2. Whether the Board erred in determining that dependent claims 2, 9, and 16 of the '915 patent are unpatentable based upon its erroneous interpretation of the "rubberbanding" claim limitation.

STATEMENT OF CASE AND FACTS

A. Apple's '915 Patent

The '915 patent relates to software that enables different types of user inputs, such as scrolling and gesturing, on a touchscreen device. Appx61 (1:7-8). It is titled "Application Programming Interfaces For Scrolling Operations" and names two Apple engineers, Andrew Platzer and Scott Herz, as inventors. Appx22. Apple filed the application for the '915 patent on January 7, 2007. *Id.* Two days later, Apple announced the first iPhone. A few months after the '915 patent application was filed, Apple released the iPhone. Appx661 (Nieh) (¶ 13). According to an expert in touchscreen mobile devices, the invention of the '915 patent contributes to the widely-acclaimed usability and intuitiveness of the iPhone's touch-driven interface. *Id.*

1. Distinguishing between “scrolls” and “gestures”

Touchscreen devices are limited by their display size, software, processing capability, and user interfaces that make them harder to use. Appx61 (1:48-50); Appx660 (Nieh) (¶ 8). Before the invention of the ’915 patent, user interfaces for touchscreen devices had difficulty interpreting user input to do what the user intended. Appx61 (1:52-55). There was no consensus on how to interpret user input to distinguish between a scroll and a gesture. Appx660 (Nieh) (¶¶ 7-8). Due to that challenge, touch-based mobile devices that allowed both scrolling and gestures were not commercially successful. Appx660 (Nieh) (¶ 9).

The ’915 patent discloses interfaces to easily interpret, and distinguish between, scrolling and gesturing inputs on a touchscreen device. Appx61 (1:59-67). As the background section of the patent explains, “[s]crolling is the act of sliding a directional (e.g., horizontal or vertical) presentation of content, such as text, drawings, or images, across a screen or display window.” Appx61 (1:40-44). “In a typical graphical user interface, scrolling is done with the help of a scrollbar or using keyboard shortcuts often the arrow keys.” Appx61 (1:42-43). In a touchscreen device as claimed in the ’915 patent, “a single touch that drags a distance across a display of the device may be interpreted as a scroll operation.” Appx63 (6:39-41).

In contrast to scrolling, "[g]esturing is a type of user input with ***two or more*** input points." Appx61 (1:45-46).[1] As the patent explains, "[t]he gesture operations include performing a scaling transform such as a zoom in or zoom out in response to a user input having ***two or more*** input points." Appx61 (2:23-26); Appx63 (5:42-45). "The gesture operations also include performing a rotation transform to rotate an image or view in response to a user input having ***two or more*** input points." Appx61 (2:26-29); Appx63 (5:45-48).

A fundamental step in the invention, shown in Figure 1 of the patent, is to determine whether to categorize a user input as a scroll or a gesture:

[1] All emphases are added unless indicated otherwise.

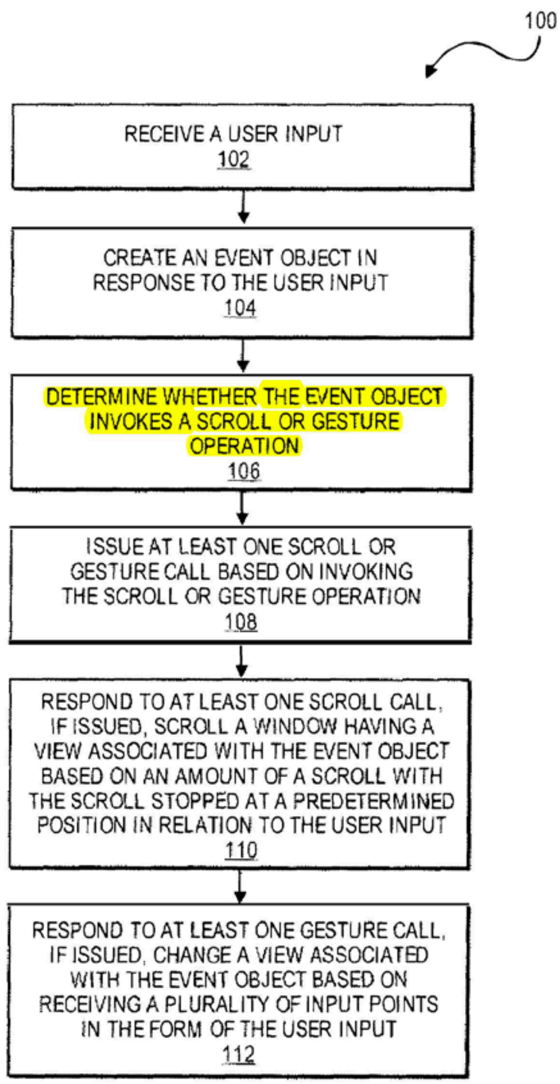


FIG. 1

Appx24 (Fig. 1) (highlighting added).

As described in the specification, the method includes determining whether an "event object" generated by a user input invokes one of two categories of operations: either "a scroll or gesture." Appx63 (6:36-39). The invention distinguishes between a scroll and a gesture with a single rule based upon the number of fingers touching the screen. "For example, a ***single*** touch that drags a

distance across a display of the device may be interpreted as a scroll operation." Appx63 (6:39-41); *see also* Appx63 (6:64-67) (describing a scroll operation with reference to "user input in the form of ***a*** mouse/finger down"). And "a ***two or more*** finger touch of the display may be interpreted as a gesture operation." Appx63 (6:41-43); *see also* Appx63-64 (6:57-60, 7:4-10) (referring to gesture operations as each receiving "a ***plurality*** of input points in the form of the user input").

The '915 patent does not describe any category of user input other than a scroll or a gesture; every user input is determined to be in one of those two categories. The patent discloses a single, consistent rule for categorizing user input, where the number of touches determines whether an input invokes ***either*** a scroll or a gesture operation. Under no circumstances in the patent can a particular number of touches sometimes indicate a scroll operation and other times indicate a gesture operation. One-finger touches are scroll operations. Two-finger, or three-finger, or four-finger touches, and so on, are gesture operations.

This categorical distinction between a scroll and a gesture operation appears in every claim of the patent. Claim 1 is directed to a method that includes the following step:

> determining whether the event object invokes a scroll or gesture operation by ***distinguishing between a single input point*** applied to the touch-sensitive display that is interpreted as the scroll operation

> ***and two or more input points*** applied to the touch-sensitive display that are interpreted as the gesture operation.

Appx72 (23:23-28). Independent claims 8 and 15 are directed to a machine readable storage medium and an apparatus that follow the same rule. Appx72 (24:5-10, 24:57-62). Every other claim depends from claim 1, 8, or 15. Every claim in the patent shares the inventors' single rule for distinguishing between a scroll and a gesture.

2. The "rubberbanding" feature

The '915 patent also discloses a way to improve scrolling on a touchscreen device using a technique the inventors called "rubberbanding." Appx64 (7:59-67). When a user scrolls past the edge of the content in a window, the scroll continues beyond that edge until it reaches a "maximum displacement value," and then "the content slides back" to fit in the window. Appx64 (7:59-67); *see also* Appx61 (2:15-21).

Four figures illustrate "rubberbanding" when a user scrolls past the end of a list of email messages. Appx64 (8:65-67) ("The rubberband operation of method 300 is illustrated in the example of FIGS. 6A-6D with the listed items being email messages."). The scroll begins in Figure 6A, when the user swipes one finger down the screen. In Figure 6B, the user reaches the top item in the list—in this example, the most recent email message, which is from Aaron Jones regarding Project Orion:

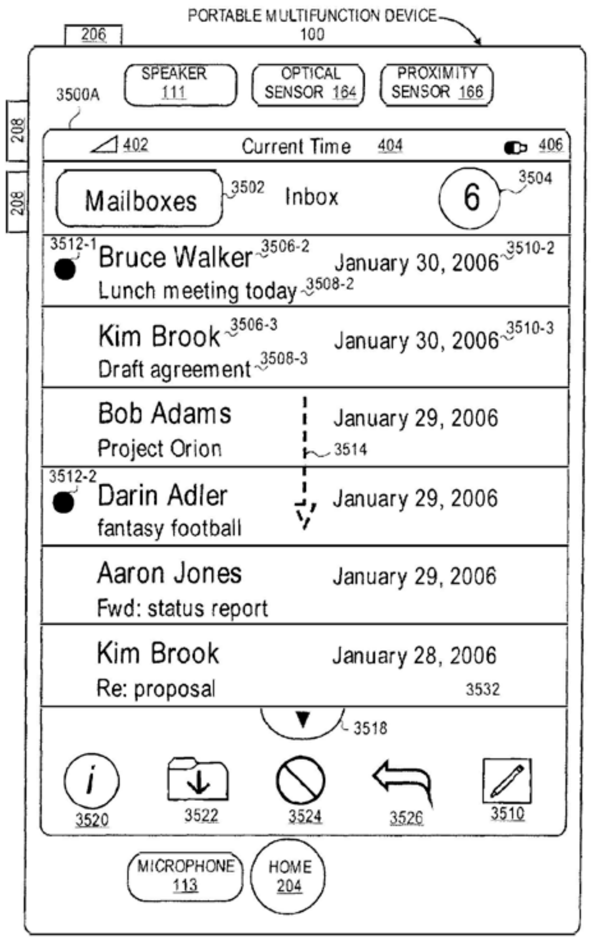


FIG. 6A

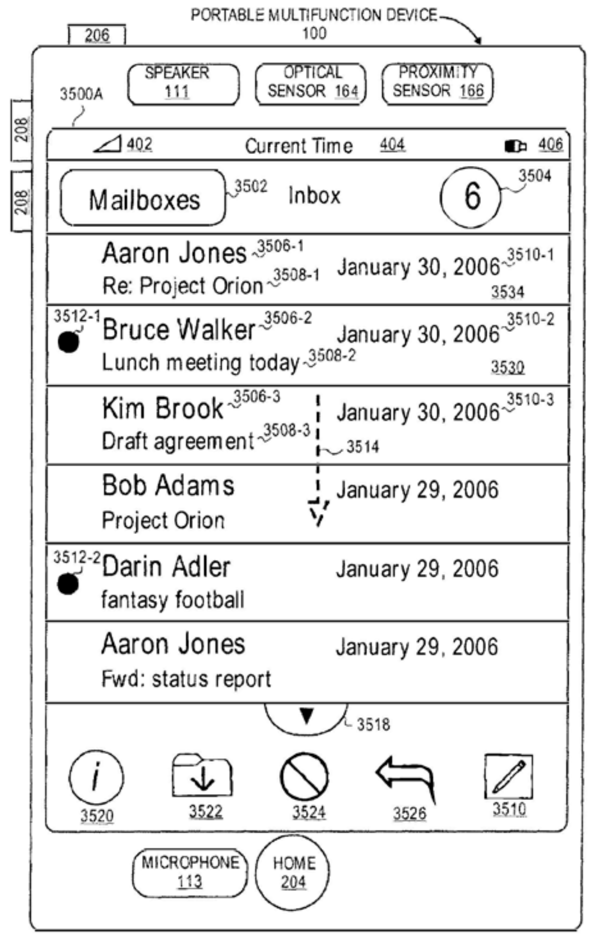


FIG. 6B

Appx30-31; *see also* Appx64-65 (8:61-9:28).

In Figure 6C, the scroll continues past the edge of the content, showing blank space (3536) above the list of email messages, until it reaches a maximum displacement value. Then, in Figure 6D, "the content slides back" (Appx64 (7:66)), showing the first message (from Aaron Jones) at the edge of the window:

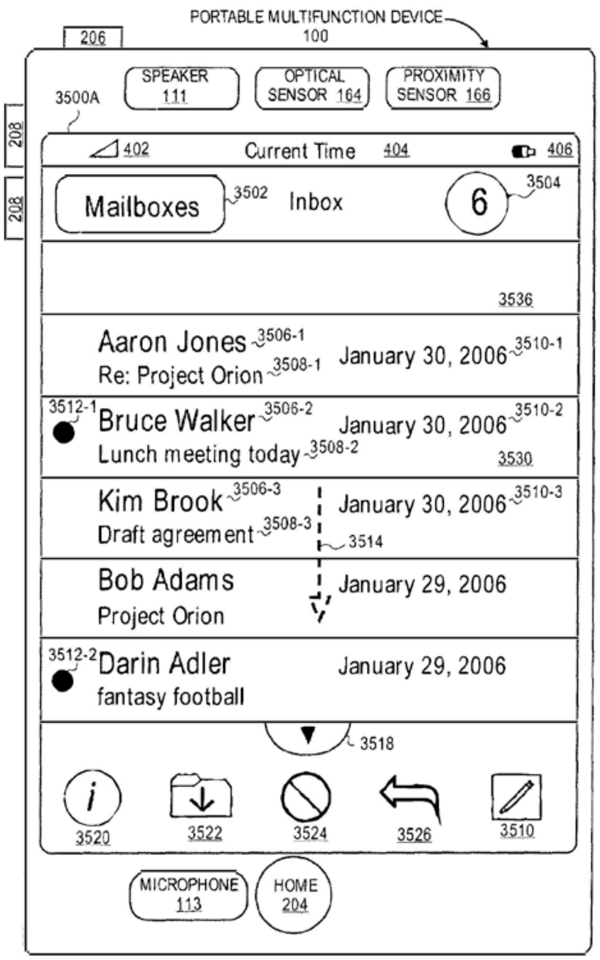


FIG. 6C

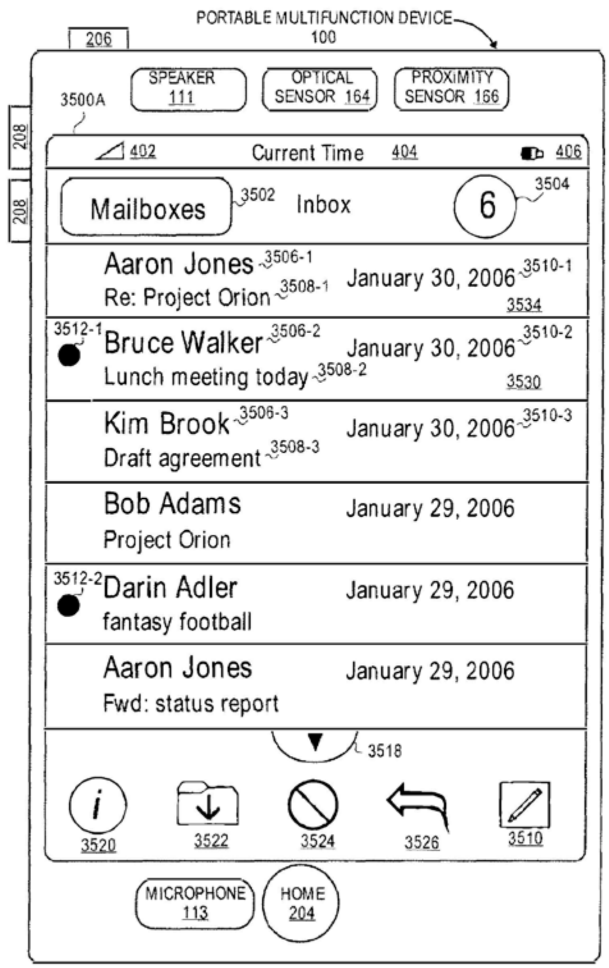


FIG. 6D

Appx32-33; *see also* Appx65 (9:29-46). As the patent explains, when the list slides back, it "is scrolled in ***an opposite direction*** until the [blank] area 3536 is no longer displayed." Appx65 (9:39-46); *see also* Appx61 (2:63-67) ("FIG. 6A-6D illustrate the scrolling of a list of items to a terminus of the list, at which point an area beyond the terminus is displayed and the list is then scrolled ***in an opposite direction*** until the area beyond the terminus is no longer displayed, in accordance with some embodiments[.]").

Dependent claims 2, 9, and 16 add this "rubberbanding" feature to the inventors' framework for implementing scrolls and gestures on a touchscreen device. Each of these claims recites:

> rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll.

Appx72-73 (23:42-46, 24:22-26, 25:5-9). As disclosed and claimed in the '915 patent, "rubberbanding" requires both scrolling beyond the window edge by a "maximum displacement value," and that "the content slides back" to fit the window. *Id.*; *see also* Appx61 (2:15-21); Appx64 (7:59-67).

B. The Prior Art

Although touchscreen devices that used scrolls and gestures were known in the prior art, they often were not intuitive or easy to use. Before the invention of the '915 patent, no one had come up with a practical way of interpreting user inputs to distinguish between scrolls and gestures without consuming significant hardware/software resources or interfering with the ease of using the device. Nor had anyone used or described the user-friendly feature of "rubberbanding" on a touchscreen device, whereby the content slides back when a scroll exceeds a threshold, before the invention of the '915 patent.

1. Hillis

U.S. Patent No. 7,724,242, filed in 2005 and naming Daniel Hillis as an inventor, is titled "Touch Driven Method And Apparatus To Integrate And Display Multiple Image Layers Forming Alternate Depictions Of Same Subject Matter." Appx525. Hillis discloses an "interactive display system," including a touch sensitive display. Appx525 (abstract). Hillis teaches that gestures can allow the user to pan, zoom, or rotate an image (Appx543 (8:4-8)), but it does not provide a rule to distinguish between scrolls and gestures based upon the number of input points. Appx664 (Nieh) (¶ 25).

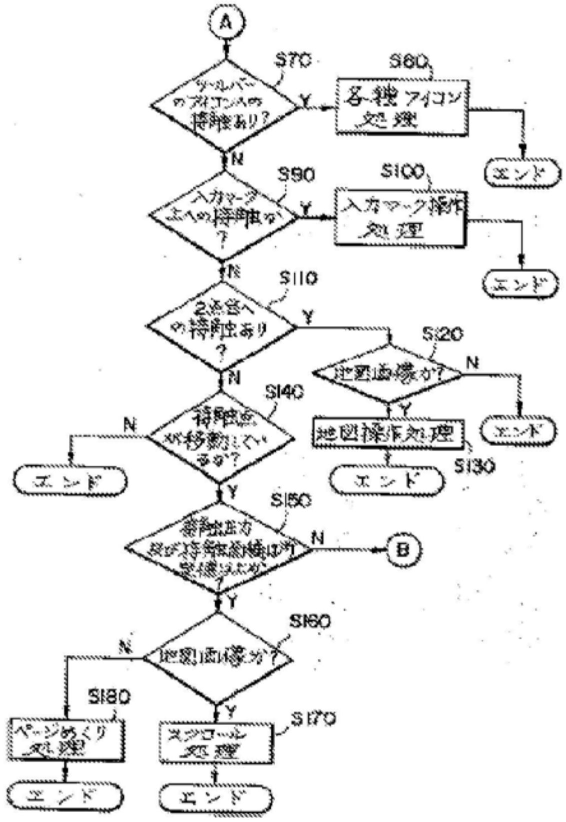
Instead, Hillis discloses using a "gesture dictionary" to identify one or more user gestures from "a predefined set of defined gestures." Appx541 (3:4-20); *see also* Appx528 (Fig. 1A). When the user touches the display screen, the system "analyzes the history of position, velocity, force, and other touch characteristics to recognize when the user has performed a recognized 'gesture.'" Appx542 (6:59-67); *see also* Appx532 (Fig. 2A). If the user input "matches a predetermined pattern" defined in the dictionary, "and therefore constitutes a 'gesture,'" the system maps the detected gesture to the action associated with that particular gesture. Appx543 (7:45-65). If the user input does not match any predefined gesture in the dictionary, then the system may "provide feedback to the user that an attempted gesture was not recognized." Appx543 (7:57-60).

2. Nomura

Japanese Patent Application No. 2000-163031 by Yasuhiro Nomura was published in 2000 and is titled "Mobile Information Device and Information Storage Media." Appx398. Nomura describes "an electronic book that includes a display area." *Id.* Nomura teaches that a user may use the touch-sensitive display area to perform operations such as scrolling, zooming in or out, and rotating an image. *Id.*

Nomura does not distinguish between scrolls and gestures based upon whether they have one input point or multiple input points. Appx661-662 (¶¶ 14-18). Instead, Nomura determines only whether or not the user input has ***precisely two*** input points, as indicated in Step 110 below:

Fig. 34



[Second column]
(A)
S70. Is there contact on an icon on the toolbar?
S90. Is there contact on the input mark?
S110. Is there contact with two items?
S140. Is the contact point moving?
S150. Is contact pressure and/or contact surface area above specified values?
S160. Is the display a map image?
S170. Scroll processing
End

Appx452 (highlighting added).

If Nomura detects exactly two input points, then the system interprets the user input as a gesture, which may be a zoom-in, zoom-out, or rotate operation. Appx424 (¶ 0159); *see also* Appx408-409 (¶ 0053) (explaining that "two fingers moving apart" is a "zoom-in operation," "two fingers moving toward each other" is

a "zoom-out operation," and "one finger rotating with another finger" is an "image rotate operation").

"If there are not two contact points," then Nomura's system by default interprets the input as a scroll operation or, depending upon other conditions, as a command to turn a page or drag an item into a folder. Appx424 (¶¶ 0160-0162); *see also* Appx408-409 (¶¶ 0053, 0056) (describing "moving one finger" as a "scroll operation"). Thus, input from greater than or fewer than two fingers would not generate the zoom and rotate commands. Appx424. Rather, Nomura interprets any input other than two contact points in exactly the same way; it does not distinguish between one input point and three, four, five, or more input points. Appx661-662 (Nieh) (¶¶ 14-18).

3. Lira

International Patent Application Publication No. WO 03/081458, to Luigi Lira and titled "Controlling Content Display," describes a method of reformatting content into columns for viewing on a small display screen. Appx461 (abstract); Appx463 (1:16-20). In one implementation, Lira provides a method for a user to scroll within a column of content and discloses a "vertical alignment control" function to "snap" the content into alignment when a user stops scrolling. Appx477 (15:18-21). The content is centered within the window when the user's scroll does not exceed a threshold, but "snaps" ***ahead*** to the next column when the

user's scroll does exceed the threshold. Appx477 (15:18-31); Appx519 (Fig. 14B). As Lira explains, scrolling past the threshold "indicates an intention to move ***beyond*** the boundary of the [centered] column." Appx477 (15:25-28).

C. Invalidity Challenge Rejected In *Samsung* Litigation

In April 2011, Apple sued Samsung for infringement of the '915 patent (and other Apple patents) in the Northern District of California. *See Apple Inc. v. Samsung Elecs. Co.*, 786 F.3d 983, 989 (Fed. Cir. 2015). Neither party asked the district court to construe the "scroll or gesture" claim limitation, agreeing instead that the term should encompass the full scope of its plain and ordinary meaning. *See Apple Inc. v. Samsung Elecs. Co.*, No. 11-cv-1846, 2012 WL 1123752, at *27-29 (N.D. Cal. Apr. 4, 2012). In fact, both Apple and Samsung agreed that the "scroll or gesture" limitation refers to "distinguishing between a single input point which is touched on the input display or multiple input points." *See Apple Inc. v. Samsung Elecs. Co.*, No. 11-cv-1846, Dkt. No. 1841, Trial Tr. 2899:23-2900:4 (N.D. Cal. Aug. 15, 2012) (Samsung expert witness Stephen Gray); *id.* at 2900:5-9 ("If it is a single input point, it's a scroll operation. If it's multiple input points, we're talking about a gesture[.]"); *accord Apple*, No. 11-cv-1846, Dkt. No. 1695, Trial Tr. 1817:22-1818:22 (Aug. 10, 2012) (Apple expert witness Karan Singh).

At trial, Samsung argued that asserted claim 8 of the '915 patent was invalid because, among other reasons, it was anticipated by the Nomura reference. *See*

Apple, 786 F.3d at 1003-1004. Dr. Karan Singh, a computer science professor at the University of Toronto and Apple's expert in the litigation, testified that the claims of the '915 patent do not cover a device that interprets one touch as a scroll, two touches as a gesture, and three touches as a scroll. Appx752 (3624:6-18). Dr. Singh also explained that none of the references cited by Samsung, including Nomura, anticipates claim 8. Appx752 (3625:10-3626:24); *see also Apple*, No. 11-cv-1846, Dkt. No. 1843, Trial Tr. 3622:19-3628:10 (Aug. 17, 2012).

In August 2012, a jury rejected Samsung's anticipation arguments based upon Nomura and other prior art, and found that claim 8 of the '915 patent is not invalid. *See Apple*, 786 F.3d at 989, 1003-1004. The jury also found that Samsung infringed claim 8 and awarded damages. *Id.* at 989-990. The district court denied Samsung's motion for judgment as a matter of law with respect to infringement, validity, and damages for the '915 patent. *See id.* at 990. This Court affirmed all aspects of the judgment relating to the '915 patent, including that claim 8 is not invalid. *See id.* at 1003-1004. In its petition for certiorari, Samsung did not raise any issues involving the '915 patent. *See Samsung Elecs. Co. v. Apple Inc.*, No. 15-777 (U.S. Dec. 14, 2015).

D. Reexamination Proceedings Before The PTO

On May 30, 2012, just two months before the *Samsung* trial, an unidentified third party filed a request for *ex parte* reexamination of claims 1-21 of the '915

patent. Appx571-572. The PTO granted that request on August 17, 2012, and issued a first office action rejecting all claims on December 19, 2012. Appx576-587; Appx588-625. The Examiner conducted an interview on March 14, 2013, and Apple filed a response to the first office action as well as a declaration from Jason Nieh, Ph.D., on March 19, 2013. Appx659-666; Appx626-658; Appx669-670.

The Examiner issued a final office action rejecting all claims on July 26, 2013. Appx671-730. After an interview with the Examiner on October 17, 2013, Apple filed a response to the final office action on October 28, 2013. Appx731-734; Appx735-750. Apple also submitted a declaration from Scott Klemmer, Ph.D., and trial testimony from Karan Singh, Ph.D. Appx751-752; Appx753-757. The Examiner issued an advisory action rejecting all claims on November 20, 2013. Appx758-767.

1. The Examiner's reasons for rejection

The Examiner's rejection of all claims relied upon two primary references, which the Examiner claimed taught the "scroll or gesture" limitation: the same Nomura patent application considered and rejected in the *Samsung* litigation, and the Hillis patent. *See* Appx676-677; Appx697-698; Appx700-701; Appx703-704; Appx713-714; Appx717; Appx721. Specifically, the Examiner rejected claims 1, 5-8, 12-15, and 19-21 as anticipated by Hillis, or as obvious over Nomura in view of a thesis by Dean Harris Rubine ("Rubine"). Appx695-696; *see also* Appx83

(Rubine). The Examiner rejected dependent claims 2, 9, and 16 as obvious over Hillis in view of Lira, or as obvious over Nomura in view of Rubine and further in view of Lira. Appx695-696. And the Examiner rejected dependent claims 3, 4, 10, 11, 17, and 18 as obvious over Hillis in view of U.S. Patent No. 6,757,673 ("Makus"), or as obvious over Nomura in view of Rubine and further in view of Makus. Appx695-696; *see also* Appx553-570 (Makus).[2]

The Examiner acknowledged that for all claims the rejection was based upon his disputed construction of the "scroll or gesture" claim limitation (Appx675-680), which requires:

> determining whether the event object invokes a scroll or gesture operation by ***distinguishing between a single input point*** applied to the touch-sensitive display that is interpreted as the scroll operation ***and two or more input points*** applied to the touch-sensitive display that are interpreted as the gesture operation.

Appx72 (23:23-28, 24:7-11, 24:58-62). The Examiner noted Apple's argument that the patent claims a rule that determines whether the user input is a scroll or a gesture by distinguishing between two categories: "(a) a single input point and (b) two or more input points." Appx676. But the Examiner concluded that Apple's

[2] Apple does not separately challenge the rejection of dependent claims 3, 4, 10, 11, 17, and 18 based upon the additional "scroll indicators" limitations. However, if the Court agrees with Apple's argument regarding the "scroll or gesture" limitation found in the independent claims, then the rejection of these dependent claims should also be reversed.

construction was "not the broadest reasonable interpretation consistent with the specification." *Id.*

Based upon the word "or" (appearing in the claim language referring to "two or more input points"), the Examiner determined that the claims were not limited to a single rule for distinguishing between scrolls and gestures, but instead extended to "***any algorithm*** in the prior art that distinguishes between (a) a single input point that is interpreted as a scroll operation and (b) two input points ***or*** more than two input points that are interpreted as a gesture operation." Appx676 (second emphasis in original); *see also* Appx761. Under the Examiner's construction, the claims would be met by an algorithm that distinguishes between a single input point and two input points (but does not distinguish between a single input point and three input points) ***or*** by an algorithm that distinguishes between a single input point and three input points (but does not distinguish between a single input point and two input points). Appx676; Appx761. In other words, the Examiner's construction permits a two-finger touch, or three-finger touch, or four-finger touch, and so on, to be interpreted as either a scroll or a gesture.

The Examiner sought to justify his interpretation of the "scroll or gesture" limitation as supported by the plain language of the claims and the specification, but relied upon an erroneous reading of the patent to do so. For example, the Examiner stated that "[t]he claims do not specify a case where the input consists of

more than two (e.g., three) input points" (Appx762), even though the claim language plainly refers to gestures involving "two or more input points." Appx72 (23:23-28, 24:7-11, 24:58-62). The Examiner also incorrectly stated that "[t]here are no examples in the specification of an input consisting of more than two (e.g., three) input points that is interpreted as a gesture operation" (Appx762), even though the specification provides numerous examples of gestures involving "two or more," or a "plurality" of, inputs. Appx61 (1:45-46); Appx63 (5:42-48, 6:41-43); Appx67 (13:36-39).

The Examiner acknowledged that Apple had presented expert declarations from Dr. Nieh and Dr. Klemmer to demonstrate that the Examiner's claim interpretation "would not be reasonable to one of ordinary skill in the art, in view of the claim as a whole and the specification." Appx762. But the Examiner deemed those declarations outweighed by his (incorrect) observation that the '915 claims and specification "do not specify a case where the input consists of more than two (e.g., three) input points" interpreted as a gesture. Appx762. That observation ignored the fact that the claims and specification specify—at least 15 times—that "two or more" touches are interpreted as a gesture. *See infra* pp. 36-37.

Applying his erroneous claim construction for the "scroll or gesture" limitation, the Examiner found the claims unpatentable in light of Nomura.

Appx695-696. The Examiner reached this conclusion even though he did not dispute that Nomura treats a three-finger touch, or four-finger touch, and so on, the ***same*** as a one-finger scroll, ***not*** as a multiple-finger gesture. Appx679-680 (Examiner acknowledging that "the processing of three or more input points in Nomura is apparently the same as the processing of a single input"); Appx827 (same). The Examiner found that Nomura's treatment of multiple-finger touches as scrolls was irrelevant, because Nomura's one-finger and two-finger touches satisfied one of the alternative rules covered by his broad construction. Appx680 (concluding that Nomura teaches the claimed matter because "two ***or*** more … includes" any rule that treats "two fingers moving apart" as a gesture); Appx827 (same).

The Examiner also found the claims unpatentable in light of Hillis, again based upon his erroneous claim construction. Appx695-696; Appx525-552. The Examiner stated that Hillis taught the "scroll or gesture" limitation, pointing to Hillis' disclosure of matching user inputs against a dictionary of predetermined patterns. Appx685-686. The Examiner also noted that Hillis describes "drawing a finger across the display surface" and "placing [two] fingertips on the display surface and moving them in an outwardly separating manner." Appx677; Appx685-686. The Examiner did not, however, identify any disclosure in Hillis of a rule for distinguishing between scrolls and gestures based upon the number of

input points. Instead, Hillis's system "analyzes the history of position, velocity, force, and other touch characteristics to recognize when the user has performed a recognized 'gesture.'" Appx542 (6:64-67).

With respect to dependent claims 2, 9, and 16, the Examiner acknowledged that neither Hillis nor Nomura discloses the claimed "rubberbanding" feature. Appx708 ("Hillis does not explicitly describe rubberbanding …."); Appx724 ("Nomura does not explicitly describe rubberbanding …."). Instead, the Examiner found those dependent claims unpatentable over the Lira reference. Appx690-691; Appx708-709; Appx724-725. The Examiner concluded that Lira meets the "rubberbanding" limitation even though Lira's response to a user's scroll beyond a threshold—"snapping" ***ahead*** to a new column of information—is the opposite of the '915 patent's requirement that "the content slides ***back***" in response to such user input. Appx64 (7:59-67).

2. The Board's decision on appeal

On appeal, the Board affirmed the Examiner's rejection of all claims. Appx1-13. The Board recognized that "[t]he principal issue" was the claim construction dispute regarding the "scroll or gesture" limitation. Appx6. Although the Board acknowledged the '915 patent's teaching that "any number of input points equal to two or more is interpreted as a gesture" (Appx8 (citing Appx63 (6:37-44))), the Board concluded that the "*broadest* interpretation consistent with

the ’915 patent disclosure does not require using [Apple’s] ‘two or more’ interpretation.” Appx8 (emphasis in original). Instead, the Board adopted the Examiner’s interpretation encompassing any algorithm in which one touch is a scroll and two touches are a gesture (even if more than two touches are interpreted as a scroll) ***or*** one touch is a scroll and more than two touches are a gesture (even if two touches are interpreted as a scroll). Appx8.

Applying this claim interpretation, the Board agreed with and adopted the Examiner’s findings that Nomura and Hillis teach the “scroll or gesture” limitation. Appx8; Appx10. Although the Board recognized that Nomura treats one touch as a scroll, two touches as a gesture, and ***three touches as a scroll***, it nevertheless found that Nomura satisfies the “scroll or gesture” limitation under the Examiner’s broad interpretation because Nomura “distinguishes between a single input point as a scroll operation and two input points as a gesture operation.” Appx8. The Board did not discuss what Hillis discloses regarding scrolls and gestures, but simply determined that Hillis rendered the claims of the ’915 patent unpatentable “[i]n view of the interpretation of the distinguishing limitation” and the Examiner’s findings. Appx10.

The Board also agreed with the Examiner’s reasons for rejecting dependent claims 2, 9, and 16. With respect to the “rubberbanding” feature, the Board noted the Examiner’s statement that “Lira describes a similar function by limiting the

user's scrolling to a predetermined threshold." Appx11-12. The Board did not address Apple's argument distinguishing Lira on the basis that, when a user scrolls beyond a threshold, Lira does not have "the content slide[] back" as in the '915 patent, but instead snaps ahead to a new column of content. *See* Appx11-12; Appx799.

3. The Board's decision on rehearing

The Board granted Apple's request for rehearing to consider Apple's arguments, but it did not modify its decision. Appx14-21. With respect to the test for distinguishing between scrolls and gestures, the Board reiterated that it found no error in the Examiner's interpretation as the "*broadest* interpretation consistent with the '915 patent disclosure." Appx16 (emphasis in original). The Board also stated again that Nomura's disclosure of interpreting one finger as a scroll and two fingers as a gesture (even though three fingers are interpreted as a scroll) satisfies the "scroll or gesture" limitation "in view of the broadest reasonable interpretation of the claim term." Appx17-18.

With respect to the "rubberbanding" claims, the Board ruled that Apple had waived the argument that "the result achieved by Lira is the ***opposite*** of the result disclosed by the '915 patent" by failing to make the argument in its appeal brief (Appx19)—even though Apple made the argument in those very words in its appeal brief. *See* Appx799 ("Lira's method therefore achieves the opposite effect

from rubberbanding."). The Board also noted that it found Apple's argument unpersuasive in view of the Examiner's findings and the reasons stated in the Board's original decision. Appx19.

SUMMARY OF THE ARGUMENT

The Board's decision affirming the rejection of claims 1-21 of the '915 patent is based upon claim construction error and should be reversed.

The '915 patent claims a single rule to distinguish between scrolls and gestures: touching the screen with one finger is interpreted as a scroll; touching the screen with "two or more" fingers is interpreted as a gesture. Every reference to scrolling or gesturing in the patent follows that rule. Scrolling is described with reference to one touch; gesturing is described with references to multiple touches. The patent never describes multiple, alternative rules—such as one touch to scroll, two touches to gesture, three touches to scroll.

The Board wrongly construed the "scroll or gesture" limitation appearing in each independent claim to encompass multiple, alternative rules because it misinterpreted the word "or." Even in reexamination proceedings where the "broadest reasonable interpretation" standard applies, each solitary word, such as "or," must be interpreted in the context of the claims and the patent as a whole. The Board, however, failed to look at the disputed claim term in the context of the entire claim language and specification, and disregarded the testimony of three

experts explaining how a person of ordinary skill in the art would understand the patented invention. In context, the Board's construction allowing for multiple, alternative rules is contrary to what the patent teaches and the claims require.

The Board's rejection of every claim depends upon its unreasonably broad construction of the "scroll or gesture" limitation. The Board itself acknowledged that its disputed construction was the "principal issue" in its decision. Appx6. Because the Board applied the wrong construction, its findings of anticipation and obviousness should be reversed; indeed, the Nomura and Hillis references do not disclose the single rule required by the "scroll or gesture" limitation under the correct claim construction. At a minimum, the Board's rejections should be vacated and remanded so that the PTO can apply the correct construction.

The Board's rejection of the dependent claims that require "rubberbanding" is similarly based upon a claim construction error. The Board misconstrued the '915 patent's "rubberbanding" limitation to encompass any scroll that "snaps" in any manner after crossing a threshold, and wrongly disregarded the patent's definition of "rubberbanding" which requires that "the content slide[] back." Appx64 (7:66). The content of the prior art is not disputed. Apple, the Examiner, and the Board agree that the Lira reference discloses a scroll operation that "snaps" ***ahead*** to the next column of content if a user scrolls past a threshold—the opposite of the claimed "rubberbanding" feature. The Board's rejection of dependent

claims 2, 9, and 16 should accordingly be reversed, or at least vacated and remanded, on this additional basis.

STANDARD OF REVIEW

This Court reviews "the Board's ultimate claim constructions de novo and its underlying factual determinations involving extrinsic evidence for substantial evidence." *In re Man Machine Interface Techs. LLC*, No. 2015-1562, __ F.3d __, 2016 WL 1567181, at *2 (Fed. Cir. Apr. 19, 2016). In an appeal from a reexamination, this Court determines "the broadest reasonable interpretation" of the claims "consistent with the specification." *In re Abbott Diabetes Care, Inc.*, 696 F.3d 1142, 1148 (2012). "The protocol of giving claims their broadest reasonable interpretation during examination does not include giving claims a legally incorrect interpretation." *In re Skvorecz*, 580 F.3d 1262, 1267 (Fed. Cir. 2009). "The broadest-construction rubric … does not give the PTO an unfettered license to interpret claims to embrace anything remotely related to the claimed invention." *In re Suitco Surface, Inc.*, 603 F.3d 1255, 1260 (Fed. Cir. 2010). Rather, the "claim language should be read in light of the specification as it would be interpreted by one of ordinary skill in the art." *Id*.

"Anticipation is a question of fact," *In re Suitco Surface, Inc.*, 603 F.3d at 1259, while "[o]bviousness is a question of law with underlying issues of fact," *PPC Broadband, Inc. v. Corning Optical Communications RF, LLC*, 815 F.3d 747,

751 (Fed. Cir. 2016). This Court reviews the Board's legal decisions de novo and its underlying factual determinations for substantial evidence. *Id.* "Substantial evidence is something less than the weight of the evidence but more than a mere scintilla of evidence, and means such relevant evidence as a reasonable mind might accept as adequate to support a conclusion." *In re Suitco Surface, Inc.*, 603 F.3d at 1259 (citations and internal quotation marks omitted).

The Board's finding that an argument is waived is reviewed for abuse of discretion. *Cf. Singh v. Brake*, 317 F.3d 1334, 1339 (Fed. Cir. 2003) (reviewing for abuse of discretion Board's application of rule against new arguments in interferences, 37 C.F.R. § 1.655(b)).

ARGUMENT

I. THE BOARD'S DECISION SHOULD BE REVERSED FOR ALL CLAIMS BECAUSE IT IS BASED UPON AN UNREASONABLY BROAD CONSTRUCTION OF THE "SCROLL OR GESTURE" LIMITATION.

Each claim of the '915 patent states a single rule for determining whether an event object invokes a scroll or a gesture: one input point is a scroll, and "two or more" input points are a gesture. That is the broadest reasonable interpretation consistent with the claim language and the specification, and it is supported by the testimony of all three experts who explained how a person of ordinary skill in the art would understand the claims. By contrast, the Board's interpretation of the

"scroll or gesture" limitation encompasses multiple, alternative rules in a way that is not supported by the claim language or the specification, and it is unreasonable.

Under the correct claim construction, the prior art fails to teach the "scroll or gesture" limitation found in every claim. Accordingly, because the Board's claim construction error affected its entire analysis, the Board's rejection of claims 1-21 should be reversed or, at a minimum, vacated and remanded so that the PTO may apply the correct claim construction.

A. The Patent Claims A Single Rule To Distinguish Between Scrolls And Gestures.

1. The claim language as a whole requires a single rule.

The '915 patent claims a single rule for distinguishing gestures from scrolls. Each claim requires:

> determining whether the event object invokes a scroll or gesture operation by ***distinguishing between a single input point*** applied to the touch-sensitive display that is interpreted as the scroll operation ***and two or more input points*** applied to the touch-sensitive display that are interpreted as the gesture operation.

Appx72 (23:23-28, 24:6-11, 24:57-62). The language is clear. Every user input generates an "event object," and every event object is determined to be in one of two categories: a "single" input point is interpreted as "the scroll operation," whereas "two or more" input points are interpreted as "the gesture operation." Each operation is defined in the claim language.

The structure of the claim language also shows how the invention determines whether an event object invokes a scroll or gesture operation—namely, by distinguishing between two types of input. In particular, the claim states that the invention "distinguish[es] ***between*** [1] a single input point applied to the touch-sensitive display that is interpreted as the scroll operation ***and*** [2] two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation." Appx72 (23:23-28, 24:6-11, 24:57-62). This structure of distinguishing "between [1] and [2]" further indicates that the claim divides all event objects into precisely two categories, where those categories are defined as described above: a "single" input point is a scroll operation, and "two or more" input points are a gesture operation. There is nothing in the claim language that allows a "single" input point to be interpreted as a gesture, or "two or more" input points to be interpreted as a scroll. Such a reading would eviscerate the dichotomy established by the "distinguishing between [1] and [2]" structure of the claims. "The construction that stays true to the claim language and most naturally aligns with the patent's description of the invention will be, in the end, the correct construction." *In re Papst Licensing Digital Camera Patent Litig.*, 778 F.3d 1255, 1261 (Fed. Cir. 2015) (citing *Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1250 (Fed. Cir. 1998), *adopted by Phillips v. AWH Corp.*, 415 F.3d 1303, 1316 (Fed. Cir. 2005) (en banc)); *see also ACTV, Inc. v. Walt Disney Co.*,

346 F.3d 1082, 1088 (Fed. Cir. 2003) ("[T]he context of the surrounding words of the claim also must be considered in determining the ordinary and customary meaning of those terms.").

Contrary to the Examiner's and the Board's interpretation, the claims do not use the word "or" to expand the number of rules that can be used to distinguish between scrolls and gestures. Instead, the claims articulate a single rule and use the word "or" within that rule because a user has the alternative of touching the display with "two or more" fingers to make a gesture. The word "or" identifies multiple touches that are interpreted as a gesture (two inputs, three inputs, four inputs, etc.), but it does not provide for multiple rules that may be used to distinguish between a scroll and a gesture. As stated in the claims, all touches having "two or more" input points are gestures. The claims do not allow for two-finger touches or three-finger touches to be sometimes classified as gestures and sometimes classified as scrolls.

2. The specification as a whole teaches a single rule.

"Above all, the broadest reasonable interpretation must be reasonable in light of the claims and specification." *PPC Broadband, Inc.*, 815 F.3d at 755. The construction "cannot be divorced from the specification and the record evidence." *In re Man Machine Interface Techs., LLC,* 2016 WL 1567181, at *2. In the '915 patent, the claim language requiring a single rule for categorizing events based

upon whether there is a single input or whether there are "two or more" inputs "is only reinforced by the specification." *See In re Abbott Diabetes Care Inc.*, 696 F.3d at 1149.

Throughout the specification, the patent consistently describes scrolling with reference to a single touch, and only a single touch. In describing how the invention determines whether an event object is a scroll, the patent states: "For example, ***a single touch*** that drags a distance across a display of the device may be interpreted as ***a scroll operation***." Appx63 (6:39-43). The patent further describes scrolling with reference to a single finger: "[U]ser input in the form of ***a mouse/finger down*** causes the scroll indicators to be displayed …. If ***a mouse/finger up*** is then detected, the scroll indicators are faded out[.]" Appx63-64 (6:64-7:1).

Indeed, the disclosures of scrolling make sense only if the input is one touch. For example, the patent describes how to provide vertical or horizontal scrolling based upon the "angle" (singular) formed by "a user input" that is necessarily a single touch. Appx65 (9:61-10:42). Likewise, the patent describes how to provide faster or slower scrolling based upon "a user input" where "[t]he user input has a certain speed." Appx65 (10:54-56). The patent also describes how to set a distance threshold for scrolling that requires "a user input in the form of a drag over a certain distance," again assuming one touch. Appx66 (11:5-6). A multiple-

touch scroll, with potentially multiple angles, multiple speeds, and multiple distances, would be contrary to the disclosure of the ’915 patent.

At the same time, the patent consistently describes gestures with reference to multiple touches. In addition to the claim language that is in dispute, later elements of the same claims require the gesture of zooming in or out “based on receiving ***the two or more*** input points.” Appx72-73 (23:38-40, 24:18-20, 25:1-3). Those claim elements also reinforce that the patent treats “two or more” as an atomic unit in defining the inputs that are categorized as gestures in the claimed rule, as indicated by the word “the” preceding “two or more input points.” *Id.* And three dependent claims (claims 6, 13, and 20) require the gesture of rotating a view “based on receiving a ***plurality*** of input points.” Appx72-73 (23:55-59, 24:37-41, 26:5-9).

The specification also always refers to multiple touches in the context of gestures. In the background section, the patent teaches that “[g]esturing is a type of user input with ***two or more*** input points.” Appx61 (1:45-46). The summary of the invention similarly describes “two or more” inputs as corresponding to gestures:

> The gesture operations include performing a scaling transform such as a zoom in or zoom out in response to a user input having ***two or more*** input points. The gesture operations also include performing a rotation transform to rotate an image or view in response to a user input having ***two or more*** input points.

Appx61 (2:23-29); *see also* Appx63 (5:42-48) (same). And in describing how the invention "determin[es] whether the event object invokes a scroll or gesture operation," the patent explains: "In one embodiment, a ***two or more*** finger touch of the display may be interpreted as a gesture operation." Appx63 (6:41-43).

This treatment of "two or more" inputs, or a "plurality" of inputs, being interpreted as gestures is continued throughout the specification: "[G]esture operations include … rotating a view associated with the event object based on receiving a ***plurality*** of input points." Appx64 (7:4-7). "If ***a second finger*** is detected, the gesture methods or functions are invoked." Appx66 (12:14-15). "In certain embodiments, a user input in the form of ***two or more*** input points (e.g., fingers) moves together or apart to invoke a gesture event that performs a scaling transform on the view," i.e., zooming in or out. Appx67 (13:36-39).

Despite these teachings, the Examiner stated that "there are no examples in the specification of an input consisting of more than two (e.g., three) input points that is interpreted as a gesture operation." Appx762. That statement is simply wrong. It ignores more than a dozen examples in the claims and specification describing "two or more," or a "plurality" of, touches as a gesture. Appx61-73 (1:45-46, 2:23-29, 5:42-48, 6:41-43, 7:4-7, 13:36-39, 23:24-27, 23:38-41, 24:6-11, 24:18-21, 24:57-62, 25:1-4).

The Examiner's and the Board's erroneous construction of the "scroll or gesture" limitation allows for multiple, alternative rules where the claims would cover devices that interpret one-touch inputs as scrolls and two-touch inputs as gestures (even if three-touch inputs are determined to be scrolls), ***or*** interpret one-touch inputs as scrolls and three-touch inputs as gestures (even if two-touch inputs are determined to be scrolls). Such two-finger and three-finger scrolls are not disclosed anywhere in the patent and are contrary to the express disclosure of two-finger and multiple-finger gestures. *See, e.g.*, *In re NTP, Inc.*, 654 F.3d 1279, 1288 (Fed. Cir. 2011) ("[T]he[] broadest reasonable construction … cannot be divorced from the specification and the record evidence.").

The Board's error in this case is similar to the one this Court corrected in *In re Man Machine Interface Technologies LLC*, No. 2015-1562, __ F.3d __, 2016 WL 1567181 (Fed. Cir. Apr. 19, 2016). There, the parties disputed the construction of the term "adapted to," and this Court recognized that the phrase, in appropriate contexts, could support either a narrower or a broader meaning. *Id.* at *2. During reexamination, the Board applied the broader meaning. *Id.* This Court reversed because the term, "as used in the [patent] claims and specification, has the narrower meaning." *Id.* Likewise, the '915 claims and specification show that "distinguishing between a single input point … and two or more input points" means a single rule distinguishing singular from plural, and not any set of multiple

alternative rules. *See also Microsoft Corp. v. Proxyconn, Inc.*, 789 F.3d 1292, 1298-1300 (Fed. Cir. 2015) (reversing Board's claim constructions as inconsistent with the specification and therefore unreasonable); *In re Imes*, 778 F.3d 1250, 1252-1253 (Fed Cir. 2015) (overturning a Board claim construction as "inconsistent with the broadest reasonable interpretation in view of the specification" as contrary with consistent usage of the term in the specification).

3. A person of ordinary skill in the art would understand the single rule.

Even in a reexamination, the Board cannot apply a claim construction that contradicts the understanding of a person of ordinary skill in the art. *See In re Buszard*, 504 F.3d 1364, 1367 (Fed. Cir. 2007). In *Buszard*, after the PTO rejected claims based upon a broad construction, this Court found that a person of ordinary skill would understand the claims more narrowly in the context of the entire patent. *See id.* The Court reversed the rejection and remanded for the PTO to follow the construction that people experienced in the field would understand. *Id.*

Here, Apple presented declarations and testimony from three experts in the field of the invention. Each expert discussed how a person of ordinary skill in the art would understand the "scroll or gesture" claim limitation in the context of the '915 patent. Dr. Scott Klemmer, a computer science professor at the University of California at San Diego, explained that a skilled person would understand that the patent claims "an algorithm (i.e., a set of logical instructions) that a machine would

employ to perform a determination of whether an event object, created based on a user's touch input, should invoke a scroll or a gesture operation." Appx754 (¶ 7). He further explained that, in a touchscreen device, "the algorithm of the distinguishing limitation would be used to disambiguate user input so that the input could be interpreted as a scroll operation if one condition is met (there is one input point) and interpreted as a gesture operation if a second condition is met (there are two or more input points)." Appx755 (¶ 8); *see also* Appx755 (¶ 9) (describing as "a logical test"). Dr. Klemmer specifically rejected the contention that anyone skilled in the field would understand the patent to claim multiple "alternative options" for how to distinguish gestures from scrolls. Appx756 (¶ 16); *see also* Appx755-757 (¶¶ 10-20).

Dr. Jason Nieh agreed. At the time of the invention in 2007, Dr. Nieh was a professor of computer science at Columbia University and was researching touchscreen devices. Appx660 (¶ 6). Dr. Nieh explained that the industry faced a challenge in how to distinguish user inputs such as scrolling and gesturing, and that the '915 patent provided a solution to that challenge: "If the touch input is determined to be a single input point, the event object invokes a scroll operation. If the touch input is determined to be two or more input points, the event object invokes a gesture operation." Appx661 (¶ 11).

Dr. Karan Singh, a computer science professor at the University of Toronto, testified at a trial involving the ’915 patent. Applying the plain and ordinary meaning of the “scroll or gesture” limitation, he explained that the claims do not cover a device that interprets one touch as a scroll, two touches as a gesture, and three touches as a scroll. Appx752 (3624:6-18).

The Board did not analyze any of this expert testimony, but reported that the Examiner found it “unpersuasive.” Appx7. The Examiner, in turn, wrote that the experts’ testimony was outweighed by the specification because “there are no examples in the specification of a gesture operation that results from an input consisting of more than two (e.g., three input points).” Appx823. The Examiner’s point is difficult to understand. As shown above, the claims and the specification include more than a dozen examples of “two or more” inputs, or “a plurality” of inputs, indicating a gesture; and no examples of multiple inputs indicating a scroll. *See supra* pp. 36-37. Consistent with the patent’s disclosure, the expert testimony presented in the reexamination proceedings confirms that a person of ordinary skill in the art would understand the patent to claim a single rule for distinguishing between one-touch scrolls and multiple-touch gestures. The Examiner’s flawed reasons for disregarding that expert testimony, which the Board adopted, led to an unreasonably broad claim interpretation. *See, e.g.*, *In re NTP, Inc.*, 654 F.3d at 1289 (reversing Board’s “broadest reasonable interpretation” based in part upon

expert testimony explaining that a person of ordinary skill in the art would understand the term more narrowly).

B. The Board Unreasonably Broadened The Claims To Cover Multiple, Alternative Rules.

In contrast to the single rule, the Examiner and the Board construed the patent to encompass multiple, alternative schemes for scrolls and gestures. In Apple's appeal to the Board, the Examiner's Answer argued that the word "or" (appearing in the phrase "two or more") meant that the claim provided alternative algorithms and that the disputed limitation is "deemed anticipated if any of these alternatives are shown in the prior art." Appx819. The Board agreed. Appx6-8.

The Board's construction is at odds with the claim language. As described above, the claims recite a single, straightforward rule for distinguishing between scrolls and gestures: a "single" input point is interpreted as a scroll, whereas "two or more" input points are interpreted as a gesture. *See supra* pp. 32-34. The Board's construction misapplies the word "or" to rewrite the claim limitation as multiple, alternative rules that might only distinguish between a one-input scroll and a two-input gesture (even if three inputs are interpreted as a scroll); ***or*** a one-input scroll and a three-input gesture (even if two inputs are interpreted as a scroll); ***or*** a one-input scroll and a four-input gesture (even if two and three inputs are interpreted as scrolls); and so on.

The Examiner and the Board sought to justify their broad construction by pointing to two cases interpreting the word "or." Appx763; Appx6. In *Schumer v. Laboratory Computer Systems, Inc.*, 308 F.3d 1304 (Fed. Cir. 2002), this Court stated that it has "consistently interpreted the word 'or' to mean that the items in the sequence are alternatives to each other." *Id.* at 1311; *see also Brown v. 3M*, 265 F.3d 1349, 1352 (Fed. Cir. 2001) (similar). But neither case addresses, let alone alters, how the word "or" is used ***in the '915 patent***. As explained above, the claims' use of "or" does not expand the number of rules available as alternatives for distinguishing between scrolls and gestures; it means only that the category of gestures defined by the single rule includes all touches having "two or more" (i.e., two, three, four, etc.) input points.[3] *See supra* pp. 32-34; *see also Vasudevan Software, Inc. v. MicroStrategy, Inc.*, 782 F.3d 671, 680 (Fed. Cir. 2015) (recognizing that the word "or" may be interpreted disjunctively or conjunctively, depending on the context in which it is used).

The Board's interpretation encompasses a mishmash of alternatives that contradict the '915 patent's description of the invention. Under the Board's

[3] The meaning would be different if the claims were written, for example, to say "distinguishing between a single input point that is interpreted as the scroll operation and two input points that are interpreted as the gesture operation; ***or*** distinguishing between a single input point that is interpreted as the scroll operation and three input points that are interpreted as the gesture operation; ***or*** distinguishing between a single input point that is interpreted as the scroll operation and four input points that are interpreted as the gesture operation; etc."

interpretation, two inputs or three inputs may be interpreted as a scroll. But the claim language states that "a single input point … is interpreted as the scroll operation" and "two or more input points … are interpreted as the gesture operation." Appx72 (23:23-29, 24:7-11, 24:58-62). Two-input and three-input scrolls are not disclosed anywhere in the specification; as described above, the patent consistently teaches that "two or more" inputs, or a "plurality" of inputs, are interpreted as gestures. *See supra* pp. 36-37. Indeed, in summarizing the invention of the '915 patent, the Board recognized that "***any*** number of input points equal to ***two or more*** is interpreted as a gesture." Appx8. Yet that is not what the Board's claim interpretation says.

The Board's construction also leads to absurd and arbitrary results. It includes, for example, the following rules:

- one and two touches are scrolls, and three and four touches are gestures;
- one touch is a scroll, two touches is a gesture, and three and four touches are scrolls;
- odd numbers of touches (one, three, etc.) are scrolls, and even numbers of touches (two, four, etc.) are gestures.

The Board's construction allowing multiple-touch scrolls contradicts every example of two or more touches in the patent—all of which are gestures. The

specification discusses multiple-touch gestures more than a dozen times. But the patent never discloses a multiple-touch scroll, because that violates the inventors' single rule. The Board's insistence that "or" means "alternatives," without context, misconstrues the claims to deny the "ease of use" (Appx61 (1:50)) that has made the invention of the '915 patent so successful.

C. The Board's Rejections Should Be Reversed Once The "Scroll Or Gesture" Claim Construction Is Corrected.

1. The Board's rejection for all claims depends upon its unreasonably broad claim construction.

At every stage of this reexamination, it was common ground that patentability for all claims turned on the "scroll or gesture" claim construction dispute. Inventor Andrew Platzer and the Examiner discussed the claim construction in an interview in March 2013. Appx669-670. In the Final Office Action rejecting all of the claims, the Examiner noted that Apple's "arguments with respect to the independent claims are based on a particular interpretation" of the claim language setting out the rule for gestures and scrolls. Appx675. The Examiner then spent five pages analyzing the claim construction issue as the dispositive issue for the independent claims. Appx676-680. Subsequently, the Examiner conducted an interview regarding the claim construction, and Apple submitted a further response. Appx731-734; Appx737-743. Then the Examiner submitted an Advisory Action discussing the claim construction dispute.

Appx760-763. During the appeal to the Board, the Examiner submitted an Answer beginning with ten pages about the claim construction. Appx817-827. Then, in its Decision, the Board identified the claim construction dispute as "[t]he principal issue" in the reexamination. Appx6. At no point in those filings did the Examiner or the Board ever say that the dispute about claim construction was not dispositive.

Nevertheless, in its Decision On Request For Rehearing, which the Board stated "does not modify" its Decision, the Board suggested for the first time that the claim construction was irrelevant:

> [W]hether the disputed term is interpreted as "either two or more," as proposed by the Examiner, or as requiring any number of input points that is equal to or greater than two be interpreted as a gesture operation, as argued by Appellant, the cited portions in Nomura disclose scroll and gesture operations requiring one and two finger movements, respectively.

Appx17. As a result, the Board wrote, under either proposed construction, "the outcome of our analysis remains unchanged." Appx17.

The Board's late-breaking position is incorrect. The Board and the Examiner were right on all of the other occasions when they acknowledged that the claim construction dispute was "[t]he principal issue" in this reexamination. Appx6.

Although the Board's statement is limited to four sentences, it appears to suggest that either construction of the '915 claims would cover the Nomura

reference.[4] Appx17-18. Not so. As both the Examiner and the Board previously acknowledged, Nomura uses one touch for a scroll, two touches for a gesture, and three touches for a scroll. Appx678-679; Appx8. The Board can only say claim construction is irrelevant by reinterpreting Apple's construction to be the same as the PTO's. They are not the same. For the last three years of this proceeding, Apple and the PTO have disagreed over whether the '915 patent claims a single rule for distinguishing between scrolls and gestures, or a set of multiple, alternative rules. And the PTO's claim construction has consistently served as the primary—and dispositive—basis for its rejection of all claims. *See* Appx676-680; Appx685-686; Appx760-765; Appx6-8; Appx10-11; Appx16-18; Appx20.

2. Under the correct claim construction, the prior art does not disclose the "scroll or gesture" limitation.

The Examiner's rejection of all claims relied upon Nomura or Hillis as the primary reference. Appx592; Appx695-696; Appx761; *see also* Appx5-6. For all claims, the Examiner found that either Nomura or Hillis discloses the "scroll or gesture" limitation under the broad claim interpretation allowing for multiple, alternative rules. *See* Appx676-677; Appx697-698; Appx700-701; Appx703-704; Appx713-714; Appx717; Appx721; *see also* Appx8. Neither reference teaches that

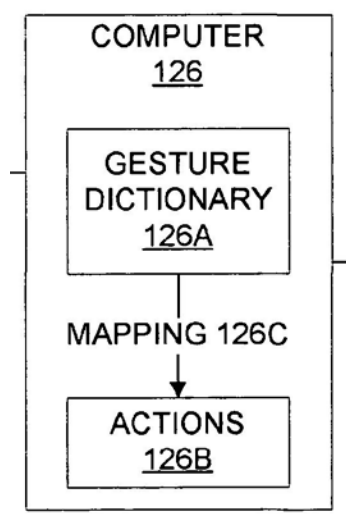
---

4 The Board did not suggest that Hillis would meet the claims under Apple's construction. *See* Appx17-18.

limitation under the correct claim interpretation requiring a single rule treating a single input as a scroll, and two or more inputs as gestures.

Under the correct construction, Nomura does not disclose the "scroll or gesture" limitation. Nomura interprets two touches as a gesture and all other touches (including one touch, three touches, four touches, etc.) as a scroll; it does not distinguish between one input and three inputs—it interprets both as scrolls. Appx424; Appx662 (Nieh) (¶¶ 16-17); *see supra* pp. 15-17. Both the Examiner and the Board agreed that this is what Nomura teaches. Appx678-679 ("The examiner generally agrees with the patent owner's analysis of the algorithm shown in Figures 33, 34 and 37 of Nomura …. The patent owner's argument that the processing of <u>three or more</u> input points in Nomura is apparently the same as the processing of a single input point is well taken." (emphasis in original)); Appx827 ("[T]he examiner generally agrees with Dr. Nieh that based on Figure 34, the processing of *three* or more input points in Nomura apparently would be the same as the processing of a single input point." (emphasis in original)); Appx8 (Board recognizing that "Nomura also uses other number of input points, such as three, for the scroll operation"). Thus, Nomura does not distinguish between one touch as a scroll and "two or more" touches as a gesture, as every claim of the '915 patent requires.

Hillis likewise fails to teach the "scroll or gesture" limitation under the correct construction. As Apple pointed out to the Board, Hillis discloses the idea of a generic gesture identification scheme. Appx793. In Hillis, gestures may allow the user to pan, zoom, or rotate an image. Appx543 (8:4-8). But Hillis does not provide a rule to determine which of those operations the user intends. Instead, Hillis discloses that its system "analyzes the history of position, velocity, force, and other touch characteristics to recognize when the user has performed a recognized 'gesture.'" Appx542 (6:64-67). Then, "the computer 126 compares the history of contact position, size, movement, velocity, and/or force" to a "dictionary 126a of predetermined gestures." Appx543 (7:51-55). Hillis does not disclose the content of that dictionary—much less the rule claimed by the '915 patent. Instead, the dictionary appears only as a box:



Appx528 (Fig. 1A).

Nor do the examples of "gestures" in Hillis disclose the "scroll or gesture" limitation claimed in the '915 patent. Hillis lists an example in which "the user has initiated a pan gesture by drawing a finger across the display" (Appx543 (8:44-46)), and another example in which a user "has gestured" to zoom in on an image "by placing his fingertips on the display surface and moving them in an outwardly separating manner" (Appx541 (3:42-45)). Many different algorithms could be used to achieve those examples: for example, gestures could be associated with two inputs, and scrolls with any other number, just as in Nomura. Or the zoom could be identified by "the history of contact position, size, movement, [and] velocity," so that two fingers moving apart are a zoom, and two fingers moving in parallel are a scroll. Hillis does not disclose any rule, let alone the single rule claimed by the '915 patent, for distinguishing between scrolls and gestures.

Thus, if the Court reverses the claim construction, the Board's rejections for claims 1-21 should also be reversed because no reference teaches the "scroll or gesture" limitation. *See, e.g.*, *In re Skvorecz*, 580 F.3d at 1268 (reversing rejection where anticipation could not be found under the correct construction as a matter of law); *In re Man Machine Interface Techs., LLC*, 2016 WL 1567181, at *4 (same). At a minimum, the Board's rejections should be vacated and remanded for the PTO to consider the prior art under the correct claim construction. *See, e.g., In re Abbott Diabetes Care, Inc.*, 696 F.3d at 1150 (vacating reexamination decision

based on incorrect construction); *In re Suitco Surface, Inc.*, 603 F.3d at 1261 (same).

II. THE BOARD'S DECISION SHOULD BE REVERSED FOR DEPENDENT CLAIMS 2, 9, AND 16 BECAUSE IT IS BASED UPON AN ERRONEOUS CONSTRUCTION OF THE "RUBBERBANDING" LIMITATION.

A. "Rubberbanding" In The '915 Patent Requires That The Scrolled Content *Slide Back* When The Scroll Exceeds A Threshold.

Three dependent claims provide a scrolling function that the inventors call "rubberbanding." Claims 2, 9, and 16 each recite:

> rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll.

Appx72-73 (23:42-46, 24:22-26, 25:5-9).

As its name suggests, "rubberbanding" involves content moving back into position after it has been stretched beyond a threshold and then released. The specification explains that this is what the inventors meant by "rubberbanding":

> Rubberbanding a scrolled region according to the method 300 occurs by a predetermined maximum displacement value when the scrolled region exceeds a display edge of a display of a device based on the scroll. If a user scrolls content of the display making a region past the edge of the content visible in the display, then the displacement value limits the maximum amount for the region outside the content. At the end of the scroll, ***the content slides back*** making the region outside of the content no longer visible on the display.

Appx64 (7:59-67). The key disclosure is the last sentence: "the content slides back" at the end of the scroll.

The patent shows the content sliding back after exceeding a threshold in the illustration of rubberbanding in Figures 6A-D. The scroll begins in Figure 6A, when a user swipes one finger down the screen. In Figure 6B, the user reaches the top item in the list—in this example, the most recent email message (from Aaron Jones regarding Project Orion):

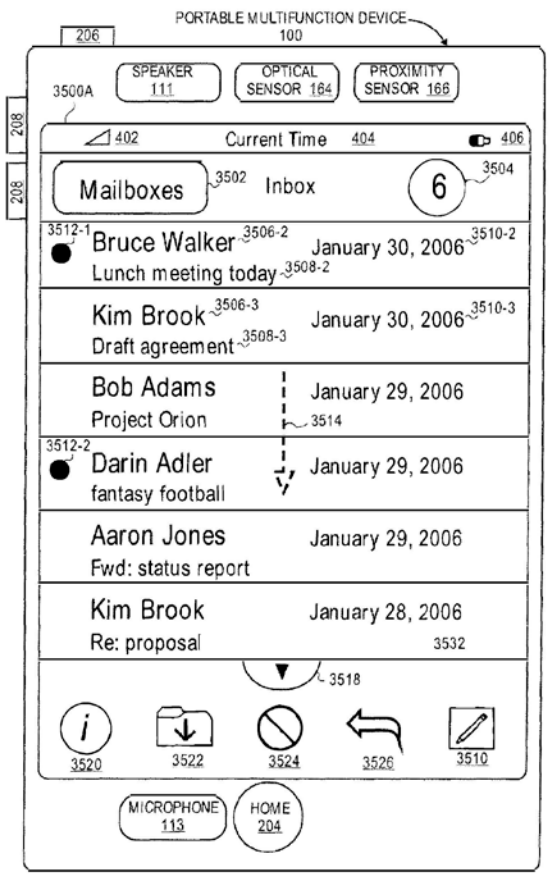


FIG. 6A

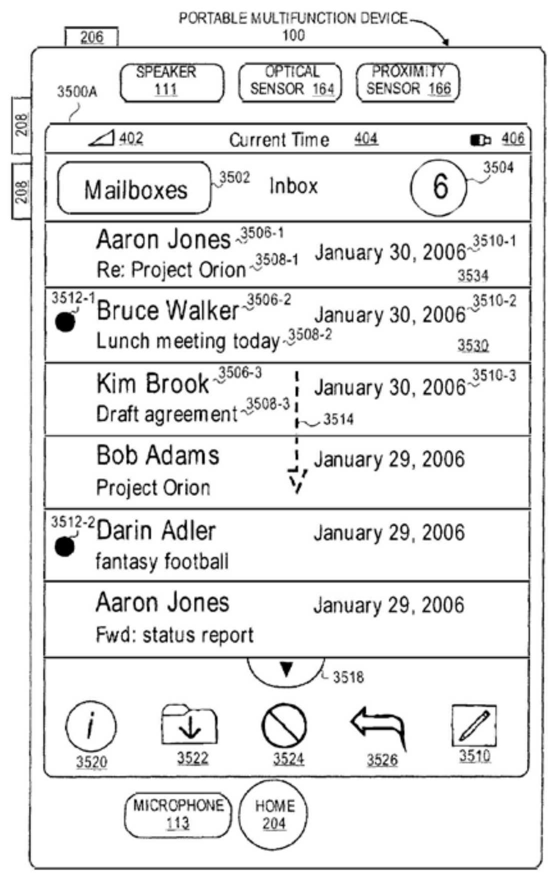


FIG. 6B

In Figure 6C, the scroll continues past the edge of the content, showing blank space above the list of email messages, until it reaches a maximum displacement value. Then, in Figure 6D, "the content slides back," showing the first message from Aaron Jones at the edge of the window. Appx64 (7:66).

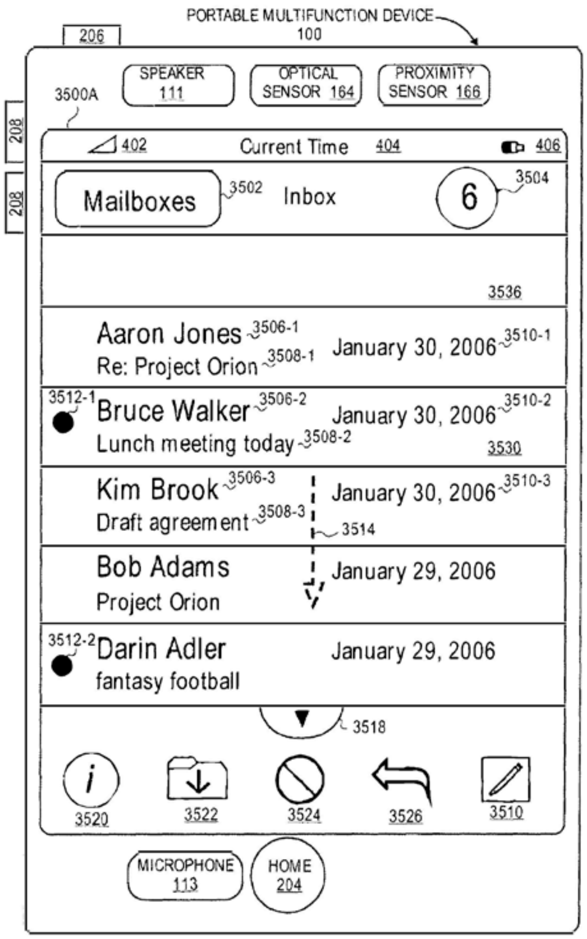


FIG. 6C

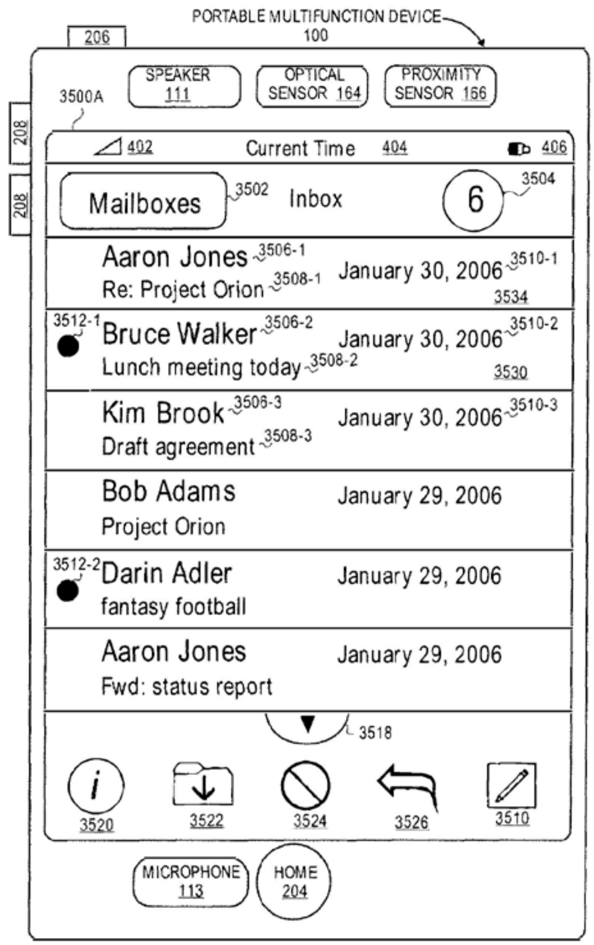


FIG. 6D

The operation is called "rubberbanding," because the scroll appears to pull the messages off the screen, exposing an empty area [3536], until it reaches a threshold and "***the content slides back***." Appx64 (7:66). As it slides back, "the list is scrolled ***in an opposite direction***" from the scroll "until the [empty] area 3536 is no longer displayed." Appx65 (9:41-42). Like a physical rubberband which has been stretched and then released, the displayed content returns to its "pre-stretched" position—compare Figures 6B, 6C, and 6D.

In describing this feature, the inventors defined "rubberbanding" as applied to scrolling across a touchscreen display in a way that is consistent with the ordinary meaning of the term in other contexts. They were clear that, after the scroll exceeds a threshold, the content slides back on the display screen. *See, e.g.*, *Phillips*, 415 F.3d at 1316 (explaining that a patent "may reveal a special definition given to a claim term by the patentee" and, "[i]n such cases, the inventor's lexicography governs"). This is the only interpretation supported by the specification. *See, e.g.*, *In re Abbott Diabetes Care, Inc.*, 696 F.3d at 1150 (construing term to match how the specification "repeatedly, consistently, and exclusively" described the claimed feature); *Interval Licensing LLC v. AOL, Inc.*, 766 F.3d 1364, 1375-1376 (Fed. Cir. 2014) (construing term in the way the specification "consistently describe[d]" the claimed feature).

B. The Board Misconstrued "Rubberbanding" To Cover Snapping *Ahead* When A Scroll Exceeds A Threshold.

Despite the patent's clear meaning of the term, the Examiner wrongly interpreted "rubberbanding" to cover snapping ***ahead*** when a scroll exceeds a threshold. Appx690. The Examiner found that the Lira reference "teaches snapping or 'rubberbanding,'" which he treated as the same. *Id.* When the scrolling exceeds a threshold in Lira, the display window "snaps to the next nearest column." Appx691. The Examiner concluded that "Lira teaches 'rubberbanding'"

by snapping ahead to the next column of the page. *Id.* The Board adopted the Examiner's findings. Appx11-12.

C. The Board's Rejection Depends Upon Its Erroneous Construction.

The erroneous construction of "rubberbanding" to mean all "snapping" misled the Examiner and the Board to find that Lira disclosed rubberbanding when Lira discloses the ***opposite*** effect. There is no dispute about the content of the prior art: the Examiner, the Board, and Apple all agree that Lira snaps ahead "to the next nearest column." Appx844; Appx691; Appx11-12. Lira does not teach "rubberbanding," when a scroll exceeds a threshold, where "the content slides back" (Appx64 (7:66)) "in an opposite direction" from the way the user swiped (Appx65 (9:41-42)). To the extent Lira "centers" the content of a column, that response occurs only when the user's scroll ***fails*** to exceed a threshold. Appx477 (15:23-25).

Lira does not disclose "rubberbanding" upon exceeding a threshold under the correct construction of that term. The Board's rejection of dependent claims 2, 9, and 16 should therefore be reversed, *see In re Skvorecz*, 580 F.3d at 1268; *In re Man Machine Interface Techs., LLC*, 2016 WL 1567181, at *4, or at least vacated and remanded for further proceedings under the correct construction. *See, e.g., In re Abbott Diabetes Care, Inc.*, 696 F.3d at 1150; *In re Suitco Surface, Inc.*, 603 F.3d at 1261.

D. Apple Did Not Waive Its Argument.

In its ruling on Apple's Request for Rehearing, the Board stated that Apple had waived its challenge to the "rubberbanding" error by failing to raise the issue in its Appeal Brief to the Board. Appx19. That waiver finding was an abuse of discretion. As the Board summarized Apple's argument on rehearing, when the scrolling threshold is met, "the result achieved by Lira is the *opposite* of the result disclosed by the '915 patent." *Id.* (emphasis in original). That is the ***very same*** argument that Apple made in its Appeal Brief to the Board. *See* Appx798-799 ("Lira's method therefore achieves the opposite effect from rubberbanding."). It is also the same argument that Apple raises here. Indeed, as Apple has maintained throughout these proceedings, the '915 patent shows and tells what "rubberbanding" means and it was improper for the Board to redefine the claims to mean its opposite. There was no waiver.

III. THE SUPREME COURT'S RULING IN *CUOZZO* MAY AFFECT THIS CASE.

The Supreme Court's decision in *Cuozzo Speed Technologies, LLC v. Lee*, No. 15-446, could affect this case. Although the question presented in *Cuozzo* is whether "the Board may construe claims in an issued patent according to their broadest reasonable interpretation rather than their plain and ordinary meaning" during *inter partes* review proceedings, the Supreme Court's decision could address or call into question whether the "broadest reasonable interpretation" is the

appropriate claim construction standard for a previously issued patent during reexamination proceedings. The Supreme Court heard argument on April 25, 2016, and is expected to issue a decision within the next few months.

Apple reserves the right to address any change in the law regarding application of the "broadest reasonable interpretation" standard, which the Examiner and the Board relied upon and emphasized in rejecting every claim of the '915 patent. Appx676; Appx8; Appx16.

CONCLUSION

The Board's decision affirming the rejection of claims 1-21 of the '915 patent should be reversed or, in the alternative, vacated and remanded for further proceedings.

Respectfully submitted,

/s/ William F. Lee

WILLIAM F. LEE
DOMINIC E. MASSA
LAUREN B. FLETCHER
SYDENHAM B. ALEXANDER, III
WILMER CUTLER PICKERING
HALE AND DORR LLP
60 State Street
Boston, MA 02109
(617) 526-6000

RACHEL KREVANS
NATHAN B. SABRI
MORRISON & FOERSTER LLP
425 Market Street
San Francisco, CA 94105
(415) 268-7000

Attorneys for Appellant Apple Inc.

May 3, 2016

ADDENDUM

TABLE OF CONTENTS

Page(s)

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/012,332	05/30/2012	7844915	P4895USREX1/106842803600	5963

20872 7590 12/09/2014
MORRISON & FOERSTER LLP
425 MARKET STREET
SAN FRANCISCO, CA 94105-2482

EXAMINER	
YIGDALL, MICHAEL J	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
12/09/2014	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte APPLE, INC.
Patent Owner and Appellant

Appeal 2014-007899
Reexamination Control 90/012,332
Patent 7,844,915[1]
Technology Center 3900

Before MAHSHID D. SAADAT, CARL W. WHITEHEAD, JR., and JASON J. CHUNG, *Administrative Patent Judges*.

SAADAT, *Administrative Patent Judge*.

DECISION ON APPEAL

Apple, Inc. (hereinafter "Appellant"), the real party in interest of Patent 7,844,915 (referred to by Appellant as "the '915 patent"), appeals under 35 U.S.C. §§ 134(b) and 306 from the Examiner's rejection of original claims 1–21 of the '915 patent.[2] We have jurisdiction under 35 U.S.C. §§ 6(b) and 306.

We affirm.

---

[1] Issued to inventors Andrew Platzer and Scott Herz on November 30, 2010, based on Application 11/620,717, filed January 7, 2007.
[2] An oral hearing was held for this Application on November 19, 2014.

STATEMENT OF THE CASE

This Ex Parte Reexamination Proceeding

This reexamination proceeding arose from a third-party request for *ex parte* reexamination filed by Joseph J. Richetti (Request for *Ex Parte* Reexamination) on May 30, 2012.

Related Litigations

Appellant has identified the following judicial proceedings related to the '915 patent (App. Br. 5–6):

1. *Apple Inc. v. Samsung Electronics Co., Ltd., Samsung Electronics America, Inc., and Samsung Telecommunications America, LLC.,* Case No. 11-cv-1846 (N.D. Cal.).

2. *In the Matter of Certain Portable Electronic Devices & Related Software,* Investigation No. 337-TA-797 (Int'l Trade Comm'n).

3. *Apple Inc. v. HTC Corp.,* Case No. ll-ev-611 (D. Del.).

The Invention and Representative Claim on Appeal

The '915 patent relates to an environment wherein a user interface software interacts with a software application through an application programming interface (API) to implement scrolling, gesturing, and animation operations (the '915 patent, col. 1, ll. 59–67). The '915 patent achieves its stated goal by creating an event object in response to the user input and determining whether the event object invokes a scroll or gesture operation (*id.*, at col. 6, ll. 37–39). Examples disclosed in the '915 patent include interpreting a single touch dragging a distance across a display of the

device as a scroll operation and a two or more finger touch of the display as a gesture operation (*Id.* at col. 6, ll. 39–43).

Representative original claims 1–3 and 5 on appeal read as follows (with emphasis):

> 1. A machine implemented method for scrolling on a touch-sensitive display of a device comprising:
>
> receiving a user input, the user input is one or more input points applied to the touch-sensitive display that is integrated with the device;
>
> creating an event object in response to the user input;
>
> determining whether the event object invokes a scroll or gesture operation *by distinguishing between a single input point* applied to the touch-sensitive display that is interpreted as the scroll operation *and two or more input points* applied to the touch-sensitive display that are interpreted as the gesture operation;
>
> issuing at least one scroll or gesture call based on invoking the scroll or gesture operation;
>
> responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object based on an amount of a scroll with the scroll stopped at a predetermined position in relation to the user input; and
>
> responding to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.
>
> 2. The method as in claim 1, further comprising: *rubberbanding a scrolling region* displayed within the window *by a predetermined maximum displacement* when the scrolling region exceeds a window edge based on the scroll.

3. The method as in claim 1, further comprising:

attaching scroll indicators to a content edge of the window.

5. The method as in claim 1, wherein determining whether the event object invokes a scroll or gesture operation is *based on receiving a drag user input for a certain time period*.

App. Br., Claims Appendix.

Prior Art Relied Upon

Makus	US 6,757,673 B2	June 29, 2004
Hillis	US 7,724,242 B2	May 25, 2010 (filed Nov. 23, 2005)
Nomura	JP 2000-163031	June 16, 2000
Lira	WO 03/081458 A1	Oct. 2, 2003

Dean Harris Rubine (Rubine), *The Automatic Recognition of Gestures*, CMU-CS-91-202 (Dec. 1991).

The Rejections on Appeal

Claims 1, 5–8, 12–15, and 19–21 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Hillis.

Claims 2, 9, and 16 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Hillis and Lira.

Claims 3, 4, 10, 11, 17, and 18 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Hillis and Makus.

Claims 1, 5–8, 12–15, and 19–21 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Nomura and Rubine.

Claims 2, 9, and 16 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Nomura, Rubine, and Lira.

Claims 3, 4, 10, 11, 17, and 18 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Nomura, Rubine, and Makus.

ANALYSIS

I. Claim Interpretation

The principal issue before us is how to interpret the "distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation" in the "determining" limitation of claims 1, 8, and 15. Appellant contends the Examiner erred in interpreting the recited "distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation" as distinguishing between one input point from two input points (App. Br. 12). Appellant relies upon a number of declarations and argues that one of ordinary skill in the art would have understood the disputed claim limitation as a distinction between one and more than one input (App. Br. 14–17). Appellant asserts this interpretation is supported by their disclosure describing "a two or more finger touch . . . interpreted as a gesture" (App. Br. 17 (citing '915 patent, col. 6, ll. 38–43)).

The Examiner disagrees with Appellant's "two or more input points" interpretation and points out the word "or" is interpreted to mean the items "two" and "more" are alternatives (Ans. 5 (citing *Schumer v. Lab. Computer Sys. Inc.*, 308 F.3d 1304, 1311 (Fed. Cir. 2002) and *Brown v. 3M*, 265 F.3d 1349 (Fed. Cir. 2001))). The Examiner concludes the two-finger operation

in Hillis and Nomura meets the claimed limitation "two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation" (*id.*). Although the Examiner acknowledges the declaration by Dr. Nieh stating the scrolling operation of Nomura is performed by two input points whereas all other inputs result in a scroll operation, the Examiner explains that "all other inputs" includes a single input point, which meets the claim limitations (Ans. 7). Similarly, the Examiner acknowledges the declarations by Dr. Klemmer and Dr. Singh and finds their statements unpersuasive in view of the description of "two or more finger touch" in Appellant's disclosure (Ans. 8–9).

Claim Interpretation Principles

"During reexamination, as with original examination, the PTO must give claims their broadest reasonable construction consistent with the specification." *In re ICON Health & Fitness, Inc.*, 496 F.3d 1374, 1379 (Fed. Cir. 2007) (citing *In re Am. Acad. of Sci. Tech Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004)). "[T]he ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, i.e., as of the effective filing date of the patent application." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1313 (Fed. Cir. 2005) (en banc).

Discussion

The invention described in the '915 patent concerns scroll and gesture operations through an application programming interface (API) in response to a user input (col. 2, ll. 3–10 and 22–29). More particularly, the disclosed method determines whether an event object invokes a scroll or a gesture operation when "a single touch that drags a distance across a display of the

device may be interpreted as a scroll operation" and "a two or more finger touch of the display may be interpreted as a gesture operation" (col. 6, ll. 37–44). That is, any number of input points equal to two or more is interpreted as a gesture.

We agree with the Examiner and conclude that giving the claim language its *broadest* interpretation consistent with the '915 patent disclosure does not require using Appellant's asserted "two or more" interpretation. As stated by the Examiner (Ans. 10), the claimed gesture operation requiring either two or more input points is met by the two finger input disclosed by Nomura and Hillis.

II. Rejection of Claims 1, 5–8, 12–15, and 19–21 Under 35 U.S.C. § 103(a) over Nomura and Rubine

Claim 1

Appellant argues Nomura distinguishes between two input points for a gesture operation and not two input points for a scroll operation, which is different from the '915 patent's one input point and more than one input points (App. Br. 19). For the reasons discussed above, we agree with the Examiner that Nomura distinguishes between a single input point as a scroll operation and two input points as a gesture operation (Ans. 12 (citing Nomura ¶¶ 53–56)). Although Nomura also uses other number of input points, such as three, for the scroll operation, the reference nonetheless discloses using one finger as an alternative to more than three for a scroll operation as well as using two fingers for a gesture operation (Nomura ¶ 56).

Appellant further argues that Rubine does not teach the recited event object because Rubine's object oriented system works with a single path instead of the claimed "gesture calls" (App. Br. 21). The Examiner points to

different teachings in Rubine (Ans. 14) and explains the teaching value of Rubine is creating an event object for the multiple inputs taught by Nomura (Ans. 15–16). We agree with the Examiner's findings and conclusion and adopt them as our own.

Next, Appellant contends a person of ordinary skill in the art would not have been motivated to combine Nomura with Rubine because the references are concerned with different problems (App. Br. 21) and the Examiner has failed to articulate a valid rationale for the combination (App. Br. 22). However, we agree with the Examiner's findings and the stated rationale with respect to the combination of the references and adopt them as our own (*see* Ans. 16–17). We also agree with the Examiner (Ans. 18) that, other than a general conclusory statement, Appellant has not provided the factual requirements of the evidence of secondary consideration and commercial success.

Claims 8 and 12–14

Appellant contends Nomura and Rubine fail to disclose the machine readable storage medium with programming instructions that operates under all circumstances because, as stated in Dr. Klemmer's declaration (¶¶ 19–20);

> A medium storing instructions for performing ***only a portion*** of the "distinguishing" limitation (*e.g.*, interpreting one input point as a scroll operation and two input points as a gesture operation) would not reasonably meet the requirements of claim 8, which requires, *inter alia*, executable programming instructions for interpreting more than two input points (3 input points, 4 input points, etc.) as a gesture operation.

(App. Br. 23). The Examiner explains paragraph 132 of Nomura discloses a machine-readable storage medium for storing instructions causing the

processor to distinguish between one input point and two input points (Ans. 19). In view of the claim interpretation discussion above, we agree with the Examiner's findings and conclusion and adopt them as our own.

Claims 5, 12, and 19

Appellant contends the combination of Nomura and Rubine does not teach or suggest the recited step of "determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period" (App. Br. 24). The Examiner points to paragraphs 9, 10, and 193 and Figures 34 and 37 of Nomura as disclosing movement history and passage of time during which contact by the user's fingers produces a drag (Ans. 20). The Examiner further explains the scroll and zoom-out processing shown in Figure 37 require a "passage of time" thus teaching the "certain time period" limitations recited in the claims (Ans. 21). We agree with the Examiner's findings and conclusion and adopt them as our own.

III. Rejection of Claims 1, 5–8, 12–15, and 19–21 Under 35 U.S.C. § 102(e) over Hillis

Claims 1, 5–8, 12–15, and 19–21

Appellant's arguments with respect to Hillis are focused on the interpretation of the claimed distinguishing limitation (App. Br. 25). Appellant further argues that Hillis does not teach an object event (App. Br. 26), a touch-sensitive display (*id.* at 27), or a device with integrated display (*id.*). In view of the interpretation of the distinguishing limitation discussed above, and the Examiner's findings and conclusions with respect to Hillis (Ans. 21–26), we are unpersuaded by Appellant's arguments.

Claims 8 and 12–14

Appellant contends Hillis fails to disclose the machine readable storage medium with programming instructions that operates under all circumstances (App. Br. 28). The Examiner explains that the disclosure of Hillis in columns 3, 5, and 8 includes references to determining steps that meet the broadest reasonable interpretation of the claims (Ans. 27). In view of the claim interpretation discussion above, we agree with the Examiner's findings and conclusion and adopt them as our own.

Claims 5, 12, and 19

Appellant contends Hillis does not teach or suggest the recited step of "determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period" (App. Br. 29). The Examiner points to disclosure of Hillis in paragraph 7 as disclosing how the contact region moves and its shape changes over time (Ans. 28). We agree with the Examiner's findings and conclusion and adopt them as our own.

IV. Rejection of Claims 2, 9, and 16 Under 35 U.S.C. § 103(a) over Lira in combination with Hillis or Nomura and Rubine

Appellant contends Lira does not disclose the "rubberbanding" limitation and merely discusses recentering or "snapping" methods for navigating the display (App. Br. 30–31). Appellant further challenges the combination of the references with Lira because neither Hillis nor Nomura relates to the internal boundaries disclosed in Lira (App. Br. 31) and the Examiner has failed to articulate a valid rationale for the combination (App. Br. 32). The Examiner states, to the extent "rubberbanding" is disclosed in column 5 of the '915 patent, Lira describes a similar function by limiting the

user's scrolling to a predetermined threshold (Ans. 29–30). Similarly, we agree with the Examiner's findings and the stated rationale with respect to the combination of the references and adopt them as our own (*see* Ans. 31–32). We also agree with the Examiner (Ans. 32) that, other than a general conclusory statement, Appellant has not provided the factual evidence required for showing the evidence of secondary consideration and commercial success (*see* App. Br. 32–3).

V. Rejection of Claims 3, 4, 10, 11, 17, and 18 Under 35 U.S.C. § 103(a) over Makus in combination with Hillis or Nomura and Rubine

Appellant contends Makus does not disclose "attaching" scroll indicators and merely discusses scroll bars that appear when more data are included in the list (App. Br. 33). Appellant further challenges the combination of the references with Makus by stating neither Hillis nor Nomura benefits from the scroll bars disclosed in Makus (*id*. at 34) and the Examiner has failed to articulate a valid rationale for the combination (*id*. at 35). The Examiner points out that because the claims do not recite indicators appear dynamically or when they are attached, the scroll bar of Makus would have suggested attaching scroll indicators to a content or window (Ans. 33). Similarly, we agree with the Examiner's findings and the stated rationale with respect to the combination of the references and adopt them as our own (*see id*. at 33–35). We also agree with the Examiner (*id* at. 35) that, other than a general conclusory statement, Appellant has not provides the factual requirements of the evidence of secondary consideration and commercial success.

DECISION

On the record before us, we conclude that the Examiner did not err in rejecting claims 1–21 under 35 U.S.C. § 103(a).

Accordingly, the Examiner's decision that claims 1–21 are unpatentable is affirmed.

TIME PERIOD FOR RESPONSE

Requests for extensions of time in this *ex parte* reexamination proceeding are governed by 37 C.F.R. § 1.550(c). *See* 37 C.F.R. § 41.50(f).

AFFIRMED

FOR PATENT OWNER:

MORRISON & FOERSTER LLP
425 MARKET STREET
SAN FRANCISCO, CA 94105-2482

FOR THIRD PARTY REQUESTER:

JOSEPH J. RICHETTI
BRYAN CAVE LLP
1290 AVENUE OF THE AMERICAS
NEW YORK, NY 10104

mls

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/012,332	05/30/2012	7844915	P4895USREX1/106842803600	5963

20872 7590 09/24/2015
MORRISON & FOERSTER LLP
425 MARKET STREET
SAN FRANCISCO, CA 94105-2482

EXAMINER	
YIGDALL, MICHAEL J	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
09/24/2015	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte APPLE, INC.
Patent Owner and Appellant

Appeal 2014-007899
Reexamination Control 90/012,332
Patent 7,844,915
Technology Center 3900

Before MAHSHID D. SAADAT, CARL W. WHITEHEAD JR., and JASON J. CHUNG, *Administrative Patent Judges*.

SAADAT, *Administrative Patent Judge*.

DECISION ON REQUEST FOR REHEARING

Appellant Apple, Inc. (hereinafter "Appellant"), the real party in interest of Patent 7,844,915 (referred to by Appellant as "the '915 patent"), requests rehearing of the Decision on Appeal mailed December 9, 2014 ("Decision"), wherein we affirmed the Examiner's rejections of claims 1–21 of the '915 patent (*see* Decision 5–11). We refer herein to Appellant's Appeal Brief filed February 26, 2014 ("App. Br."), the Examiner's Answer mailed May 2, 2014 ("Ans."), Appellant's Reply Brief filed July 2, 2014 ("Reply Br."), and Appellant's Request for Rehearing filed February 9, 2015 ("Reh'g Req."). We reconsidered the Decision in light of Appellant's

arguments in the Request for Rehearing, but, for the reasons discussed below, we do not modify the Decision.

A request for rehearing is limited to matters overlooked or misapprehended by the panel in rendering the original decision. *See* 37 C.F.R. § 41.52. Appellant's arguments do not persuade us we overlooked or misapprehended matters in the Decision, but, for completeness, we address Appellant's arguments in the Request for Rehearing below.

CLAIM CONSTRUCTION

Appellant argued in the Appeal Brief that one of ordinary skill in the art would have understood the claim limitation of "distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation" in the "determining" limitation of claims 1, 8, and 15 as a distinction between one and more than one input (App. Br. 14–17). In the Decision, we considered Appellant's proposed construction and found no error in the Examiner's construction based on giving the claim language its *broadest* interpretation consistent with the '915 patent disclosure and finding the claimed gesture operation requires either two or more input points, similar to the two-finger input disclosed by Nomura and Hillis (Decision 5–7). In the Request for Rehearing, Appellant argues that our Decision was based on claim construction that is inconsistent with the Specification, as well as the ordinary meaning and structure of the claim (Req. Reh'g 3–10).

Appellant asserts the claim language, rather than providing "a choice between gestures of two points and gestures of more than two points,"

"establishes a dichotomy between 'scroll operation' and 'gesture operations' in which every 'event object' must be determined to be either a scroll or a gesture" (Req. Reh'g 5). Appellant further challenges our Decision by arguing:

> The Board's affirmance of the Examiner's conclusion that prior art which distinguishes between a single input point and two input points--but indisputably ***does not*** distinguish between a single input point and more than two input points--misapprehends the dichotomous nature of the claim language and is contrary to Federal Circuit precedent. The claims cannot be met by prior art that practices only a portion of the "determining" step, but never performs the entire step.

(Req. Reh'g 5–6).

Appellant's argument is not persuasive. In fact, this argument, similar to Appellant's reliance on *Ferguson Beauregard/Logic Controls Div. of Dover Res. Inc. v. Mega Sys. LLC*, 350 F.3d 1327, 1346 (Fed. Cir. 2003) (*id.*), improperly focuses on whether prior art meets all the possible number of input points that could be interpreted as a scroll operation.

Although Appellant urges the term "two or more input points" is not to be interpreted as either two input points or more than two input points, the outcome of our analysis remains unchanged. That is, whether the disputed term is interpreted as "either two or more," as proposed by the Examiner, or as requiring any number of input points that is equal to or greater than two be interpreted as a gesture operation, as argued by Appellant, the cited portions in Namura disclose scroll and gesture operations requiring one and two finger movements, respectively. Namura's disclosure in paragraphs 55 and 56 includes interpreting the event as a scroll with the movement of one finger and interpreting the event as a gesture operation with the movement

of two fingers. Both of these operations meet Appellant's claimed "determining" step in view of the broadest reasonable interpretation of the claim term, which is also consistent with Appellant's Specification.

Contrary to Appellant's contention that other passages in columns 1, 5, and 7 of '915 patent indicate that the "two or more input points" limitation should be construed as narrowly as Appellant argues (*see* Req. Reh'g 8–9), these passages and others in columns 12 and 13 describe a gesture operation merely as "a user input in the form of two or more points." In fact, the term "*a two or more* finger touch" is mentioned only once in column 6 of Appellant's Specification whereas in other instances a gesture operation is performed based on "a user input having *two or more* input points" (emphasis added).

Additionally, we are unpersuaded that the cited supplemental authority, filed April 20, 2015, supports Appellant's position regarding the conjunctive interpretation of "or" in the disputed claim limitation. We observe that Appellant's argued position differs from the decision in *Vasudevan Software, Inc. v. Microstrategy, Inc.*, No. 2014-1094 (Fed. Cir. Apr. 3, 2015) because, unlike *Vasudevan*, Appellant has not pointed to any part of the prosecution history to show that Appellant was relying on the definition based on a conjunctive interpretation of "or."

PRIOR ART REJECTIONS

Rejection of Claims 2, 9, and 16

Appellant argues our Decision misapprehended the "rubberbanding" limitation because Lira does not disclose "limiting the maximum amount for the region outside the content" (Reh'g Req. 13); "there is no 'predetermined

maximum displacement' in Lira that prevents additional scrolling once a threshold is met" (*id.*); and "when the threshold *is* met—as is required by the 'rubberbanding' limitation—the result achieved by Lira is the ***opposite*** of the result disclosed by the '915 patent" (Reh'g Req. 14). Appellant's argument is untimely and waived because Appellant does not show good cause as to why this argument was not presented in the Appeal Brief. *Ex parte Borden*, 93 USPQ2d 1473, 1474 (BPAI 2010) (informative). Further, even if Appellant's additional arguments were presented timely, which they are not, they are not persuasive in view of the Examiner's findings and conclusion (*see* Ans. 29–33) and for the reasons stated on pages 10 and 11 of our Decision.

Obviousness of Claims 5, 12, and 19 over Nomura and Rubine

Appellant argues we erred because Nomura does not teach the "drag user input" because "Nomura determines whether to invoke a scroll or gesture operation ***before*** it ever assesses input movement; that determination therefore cannot be ***based on*** a drag user input" (Reh'g Req. 15). However, Appellant's contentions in the Appeal Brief were limited to Figure 34 of Nomura and whether the determination of a scroll or gesture operation is based on "receiving a drag user input ***for a certain time period***" (App. Br. 24).

As explained in the Decision, paragraphs 9, 10, and 193 and Figures 34 and 37 of Nomura disclose considering movement history and passage of time as the user's fingers produce a drag (Decision 9). We further observe that Figure 34 of Nomura at step 110 determines whether contact is made with two items or one item, then at step 140 determines if the contact point

is moving, and lastly, at step 170 processes a scroll. Therefore, the disclosed processing a scroll operation depends on determining contact with one item, moving contact point, and passage of time, which meets the disputed claim limitation because the claim does not specify a specific order for receiving a drag user input and the determination of the duration of contact.

Anticipation of Claims 5, 12, and 19 by Hillis

Appellant argues we erred because Hillis does not teach "determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period" (Reh'g Req. 19). Appellant's argument is not persuasive. As stated in our Decision, the Examiner's findings related to the disclosure of Hillis in column 7 meet the disputed claim limitation (Decision 9–10). As found by the Examiner, Hillis discloses the recited determination step or whether the activity of the current contact constitutes a "gesture" using a position history, movement velocity, and force (*see* Ans. 28 (citing Hillis col. 7, ll. 15–25 and 46–65)). That is, Hillis determines invoking a scroll or gesture based on position history, which provides a record of contact movement and change over a certain time period.

DECISION

We grant the Request for Rehearing to the extent that we reconsidered the Decision in light of Appellant's arguments in the Request for Rehearing, but we deny the Request for Rehearing in that we do not modify the Decision.

TIME PERIOD FOR RESPONSE

Requests for extensions of time in this *ex parte* reexamination proceeding are governed by 37 C.F.R. § 1.550(c). *See* 37 C.F.R. § 41.50(f).

DENIED

FOR PATENT OWNER:

MORRISON & FOERSTER LLP
425 MARKET STREET
SAN FRANCISCO, CA 94105-2482

FOR THIRD PARTY REQUESTER:

JOSEPH J. RICHETTI
BRYAN CAVE LLP
1290 AVENUE OF THE AMERICAS
NEW YORK, NY 10104

Ssc

EXHIBIT 1

US007844915B2

(12) **United States Patent**
Platzer et al.

(10) **Patent No.: US 7,844,915 B2**
(45) **Date of Patent: Nov. 30, 2010**

(54) **APPLICATION PROGRAMMING INTERFACES FOR SCROLLING OPERATIONS**

(75) Inventors: **Andrew Platzer**, Santa Clara, CA (US); **Scott Herz**, Santa Clara, CA (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 583 days.

(21) Appl. No.: **11/620,717**

(22) Filed: **Jan. 7, 2007**

(65) **Prior Publication Data**

US 2008/0168384 A1 Jul. 10, 2008

(51) **Int. Cl.**
G06F 3/00 (2006.01)
G06F 3/033 (2006.01)
G06F 3/041 (2006.01)
G06F 3/048 (2006.01)

(52) **U.S. Cl.** **715/781**; 715/784; 715/800; 345/173

(58) **Field of Classification Search** 715/764, 715/765, 784, 786, 788, 800, 864, 866, 973, 715/974; 345/156, 157, 169, 173
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,534,893	A	7/1996	Hansen et al.	
5,903,902	A	5/1999	Orr et al.	
6,028,602	A	2/2000	Weidenfeller et al.	
6,486,896	B1	11/2002	Ubillos	
6,677,965	B1 *	1/2004	Ullmann et al.	715/786
6,741,996	B1	5/2004	Brechner et al.	
6,839,721	B2	1/2005	Schwols	
6,903,927	B2	6/2005	Anlauff	
6,957,392	B2	10/2005	Simister et al.	
6,958,749	B1 *	10/2005	Matsushita et al.	345/175
7,009,626	B2	3/2006	Anwar	
7,088,374	B2	8/2006	David et al.	
7,117,453	B2	10/2006	Drucker et al.	
7,173,623	B2	2/2007	Calkins et al.	
7,337,412	B2	2/2008	Guido et al.	
7,346,850	B2	3/2008	Swartz et al.	

(Continued)

FOREIGN PATENT DOCUMENTS

EP 1517228 3/2005

(Continued)

OTHER PUBLICATIONS

Toshiyuki Masui et al; "Elastic Graphical Interfaces for Precise Data Manipulation", 1995; ACM; pp. 143-144.*

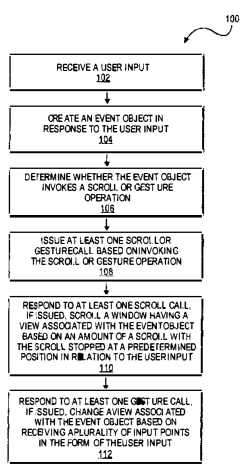
(Continued)

Primary Examiner—Xiomara L. Bautista
(74) *Attorney, Agent, or Firm*—Blakely, Sokoloff, Taylor & Zafman LLP

(57) **ABSTRACT**

At least certain embodiments of the present disclosure include an environment with user interface software interacting with a software application. A method for operating through an application programming interface (API) in this environment includes transferring a set bounce call. The method further includes setting at least one of maximum and minimum bounce values. The set bounce call causes a bounce of a scrolled region in an opposite direction of a scroll based on a region past an edge of the scrolled region being visible in a display region at the end of the scroll.

21 Claims, 37 Drawing Sheets



U.S. PATENT DOCUMENTS

7,561,159	B2	7/2009	Abel et al.	
7,576,732	B2 *	8/2009	Lii	345/173
2001/0045949	A1	11/2001	Chithambaram et al.	
2002/0194589	A1	12/2002	Cristofalo et al.	
2003/0095096	A1	5/2003	Robbin et al.	
2003/0122787	A1	7/2003	Zimmerman et al.	
2003/0132959	A1	7/2003	Simister et al.	
2003/0160832	A1	8/2003	Ridgley et al.	
2003/0174149	A1	9/2003	Fujisaki et al.	
2004/0021676	A1	2/2004	Chen et al.	
2004/0021698	A1	2/2004	Baldwin et al.	
2004/0100479	A1	5/2004	Nakano et al.	
2004/0215643	A1	10/2004	Brechner et al.	
2004/0222992	A1	11/2004	Calkins et al.	
2004/0224638	A1	11/2004	Fadell et al.	
2005/0057524	A1 *	3/2005	Hill et al.	345/173
2005/0088443	A1	4/2005	Blanco et al.	
2005/0193015	A1	9/2005	Logston et al.	
2006/0038796	A1 *	2/2006	Hinckley et al.	345/173
2006/0190833	A1	8/2006	SanGiovanni et al.	
2006/0236263	A1 *	10/2006	Bathiche et al.	715/786
2007/0055967	A1	3/2007	Poff et al.	
2007/0075965	A1	4/2007	Huppi et al.	
2007/0174257	A1	7/2007	Howard	
2007/0185876	A1	8/2007	Mendis et al.	
2007/0252821	A1 *	11/2007	Hollemans et al.	345/173
2007/0288856	A1	12/2007	Butlin et al.	
2008/0005703	A1 *	1/2008	Radivojevic et al.	715/863
2008/0016096	A1	1/2008	Wilding et al.	
2008/0034029	A1	2/2008	Fang et al.	
2008/0048978	A1 *	2/2008	Trent et al.	345/157
2008/0168395	A1 *	7/2008	Ording et al.	715/833
2008/0231610	A1 *	9/2008	Hotelling et al.	345/173
2009/0259969	A1 *	10/2009	Pallakoff	715/808

FOREIGN PATENT DOCUMENTS

GB	2 319 591	A		5/1998
GB	2319591	A	*	5/1998
WO	WO-2006/067711			6/2006
WO	WO 2008/085848	A1	*	7/2008
WO	WO 2008/085877	A1	*	7/2008
WO	WO-2008085848			7/2008
WO	WO-2008085877			7/2008

OTHER PUBLICATIONS

Office Action, U.S. Appl. No. 11/620,723, mailed Apr. 1, 2009, 8 pages.

Office Action, U.S. Appl. No. 11/620,709, mailed Apr. 1, 2009, 8 pages.

Office Action, U.S. Appl. No. 11/620,720, mailed Jun. 23, 2009, 17 pages.

Office Action, U.S. Appl. No. 11/620,720, mailed Dec. 23, 2008, 18 pages.

PCT International Search Report and Written Opinion for PCT International Appln. No. US2008/000058, mailed Jul. 31, 2008 (10 pages).

PCT International Search Report and Written Opinion for PCT International Appln. No. US2008/000089, mailed Apr. 6, 2008 (14 pages).

PCT International Search Report and Written Opinion for PCT International Appln. No. PCT/US2008/000103, mailed Jun. 3, 2008 (15 pages).

PCT International Search Report and Written Opinion for PCT International Appln. No. PCT/US2008/000069, mailed May 2, 2008 (16 pages).

PCT International Search Report and Written Opinion for PCT International Appln. No. PCT/US2008/000060, mailed Apr. 22, 2008 (12 pages).

Office Action, U.S. Appl. No. 11/620,723, mailed Jun. 8, 2010, 7 pages.

Office Action, U.S. Appl. No. 11/620,709, mailed Jun. 9, 2010, 7 pages.

Final Office Action, U.S. Appl. No. 11/620,709 mailed Nov. 13, 2009, 8 pages.

Final Office Action, U.S. Appl. No. 11/620,723, mailed Nov. 17, 2009., 10 pages.

Office Action, U.S. Appl. No. 11/620,720 mailed Nov. 18, 2009, 17 pages.

* cited by examiner

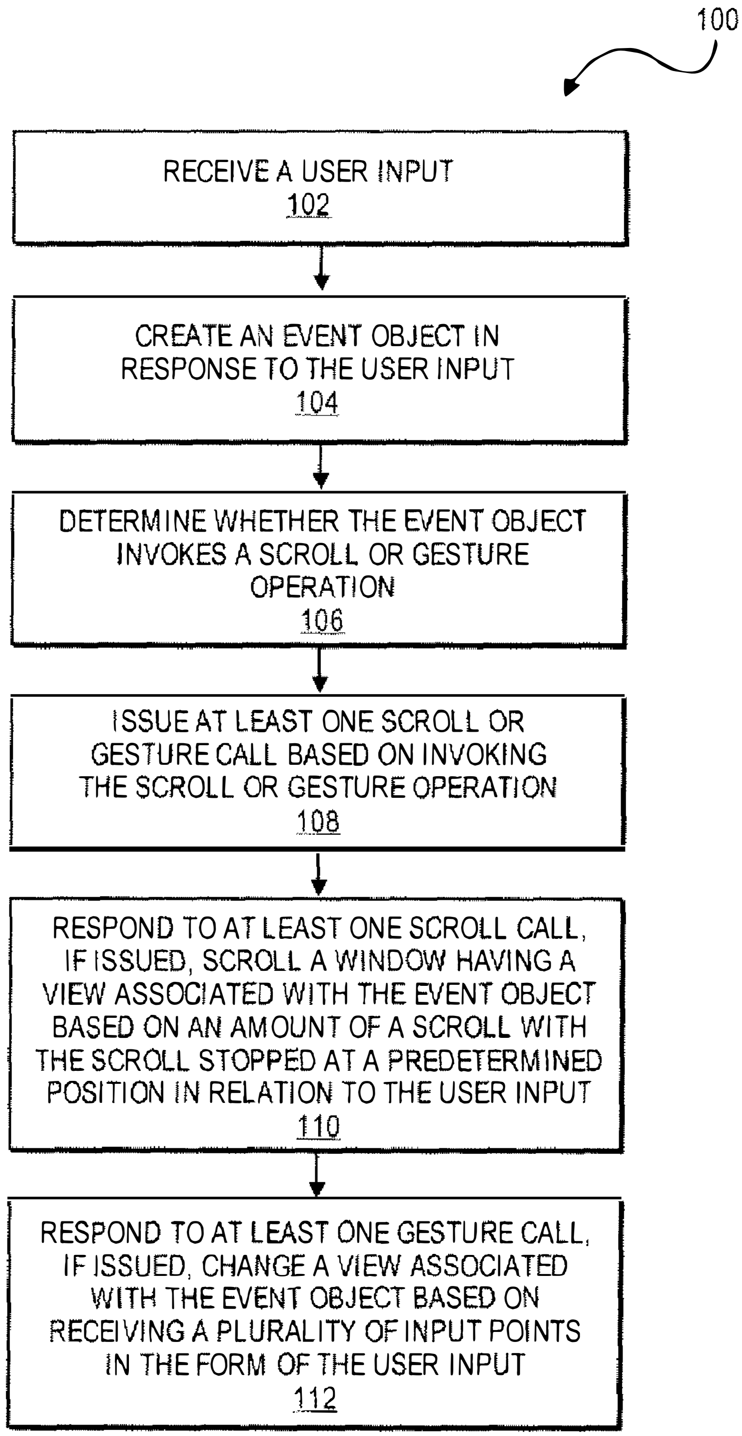

100
RECEIVE A USER INPUT
102
CREATE AN EVENT OBJECT IN RESPONSE TO THE USER INPUT
104
DETERMINE WHETHER THE EVENT OBJECT INVOKES A SCROLL OR GESTURE OPERATION
106
ISSUE AT LEAST ONE SCROLL OR GESTURE CALL BASED ON INVOKING THE SCROLL OR GESTURE OPERATION
108
RESPOND TO AT LEAST ONE SCROLL CALL, IF ISSUED, SCROLL A WINDOW HAVING A VIEW ASSOCIATED WITH THE EVENT OBJECT BASED ON AN AMOUNT OF A SCROLL WITH THE SCROLL STOPPED AT A PREDETERMINED POSITION IN RELATION TO THE USER INPUT
110
RESPOND TO AT LEAST ONE GESTURE CALL, IF ISSUED, CHANGE A VIEW ASSOCIATED WITH THE EVENT OBJECT BASED ON RECEIVING A PLURALITY OF INPUT POINTS IN THE FORM OF THE USER INPUT
112

FIG. 1

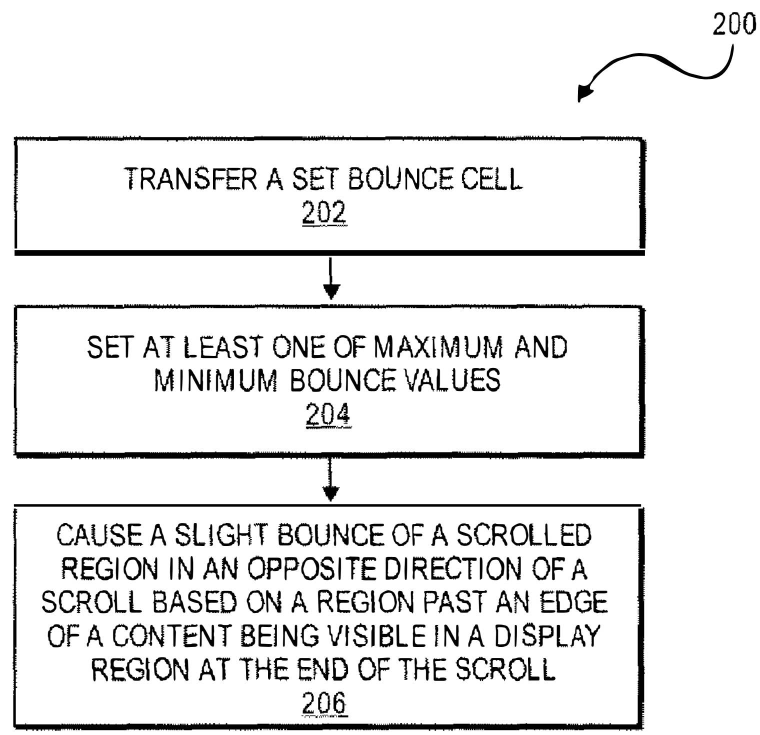

200
TRANSFER A SET BOUNCE CELL
202
SET AT LEAST ONE OF MAXIMUM AND MINIMUM BOUNCE VALUES
204
CAUSE A SLIGHT BOUNCE OF A SCROLLED REGION IN AN OPPOSITE DIRECTION OF A SCROLL BASED ON A REGION PAST AN EDGE OF A CONTENT BEING VISIBLE IN A DISPLAY REGION AT THE END OF THE SCROLL
206

FIG. 2

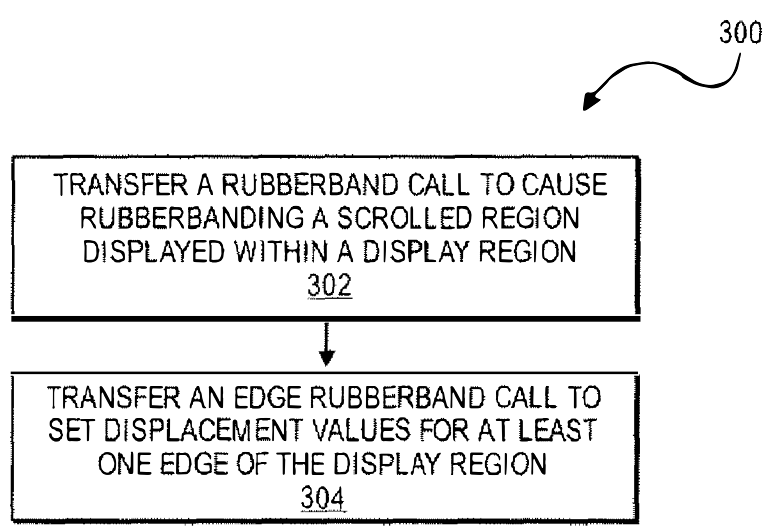

300
TRANSFER A RUBBERBAND CALL TO CAUSE RUBBERBANDING A SCROLLED REGION DISPLAYED WITHIN A DISPLAY REGION
302
TRANSFER AN EDGE RUBBERBAND CALL TO SET DISPLACEMENT VALUES FOR AT LEAST ONE EDGE OF THE DISPLAY REGION
304

FIG. 3

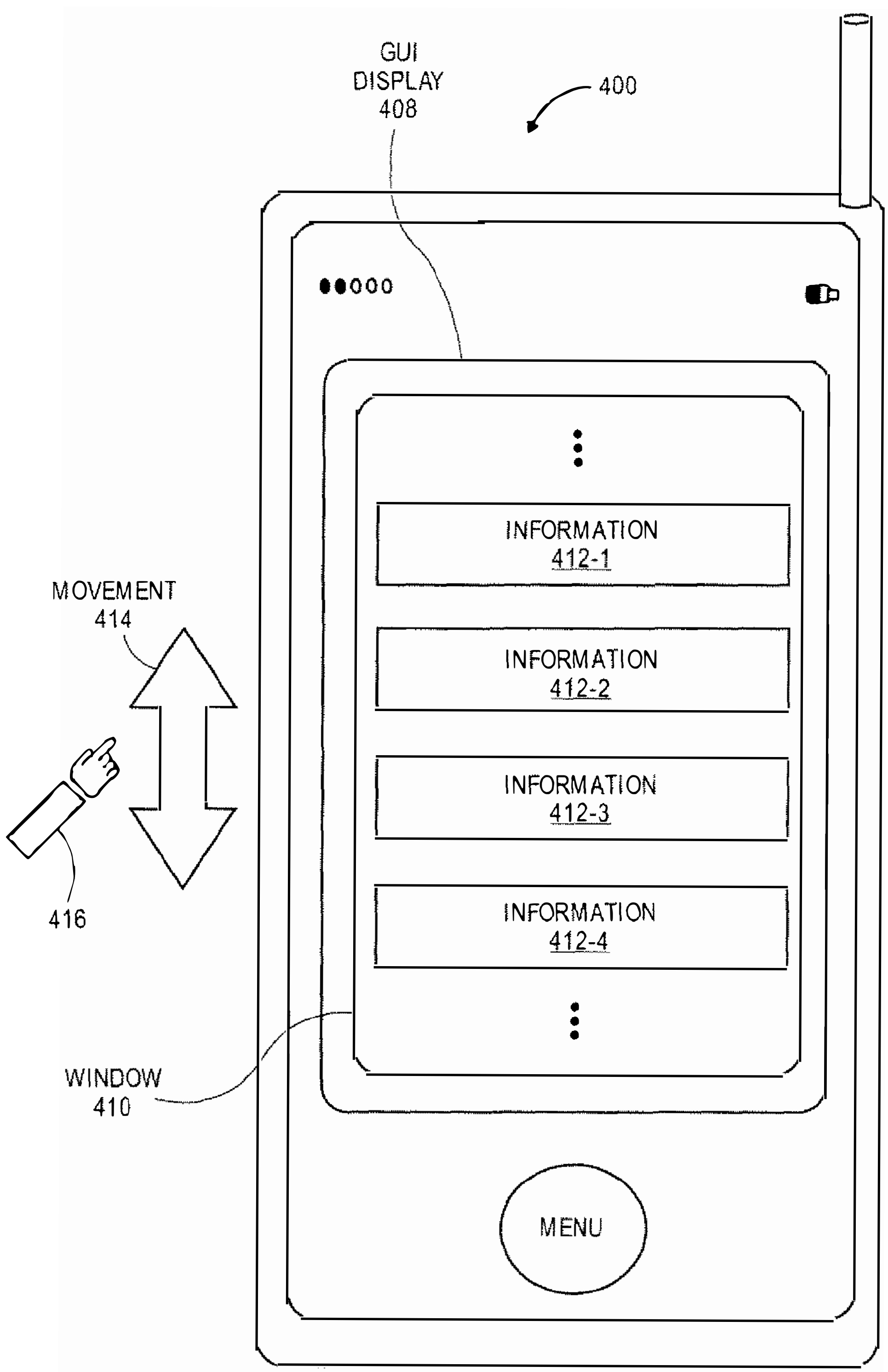

GUI
DISPLAY
408
400
INFORMATION
412-1
INFORMATION
412-2
INFORMATION
412-3
INFORMATION
412-4
MOVEMENT
414
416
WINDOW
410
MENU

FIG. 4

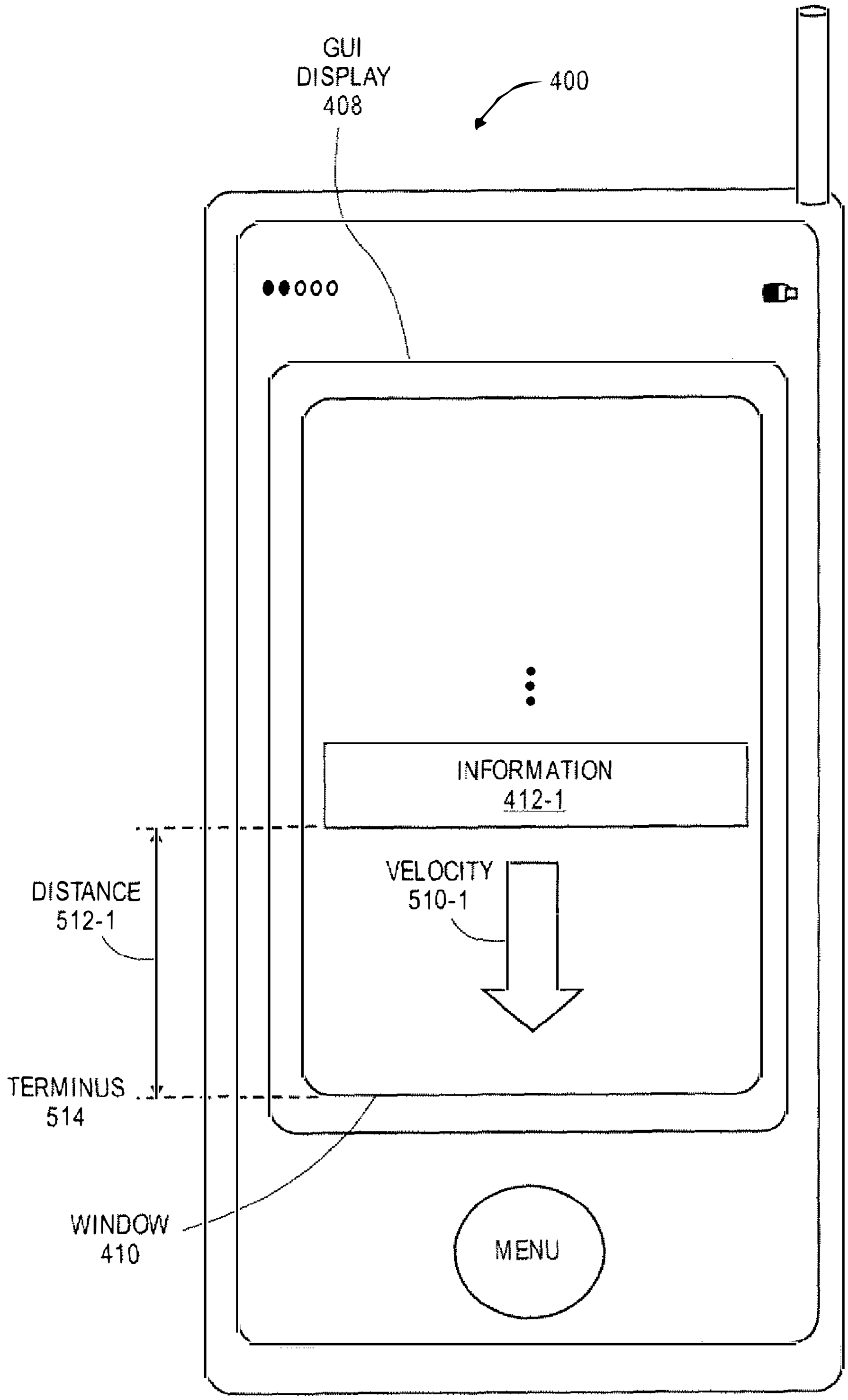

GUI
DISPLAY
408
400
INFORMATION
412-1
VELOCITY
510-1
DISTANCE
512-1
TERMINUS
514
WINDOW
410
MENU

FIG. 5A

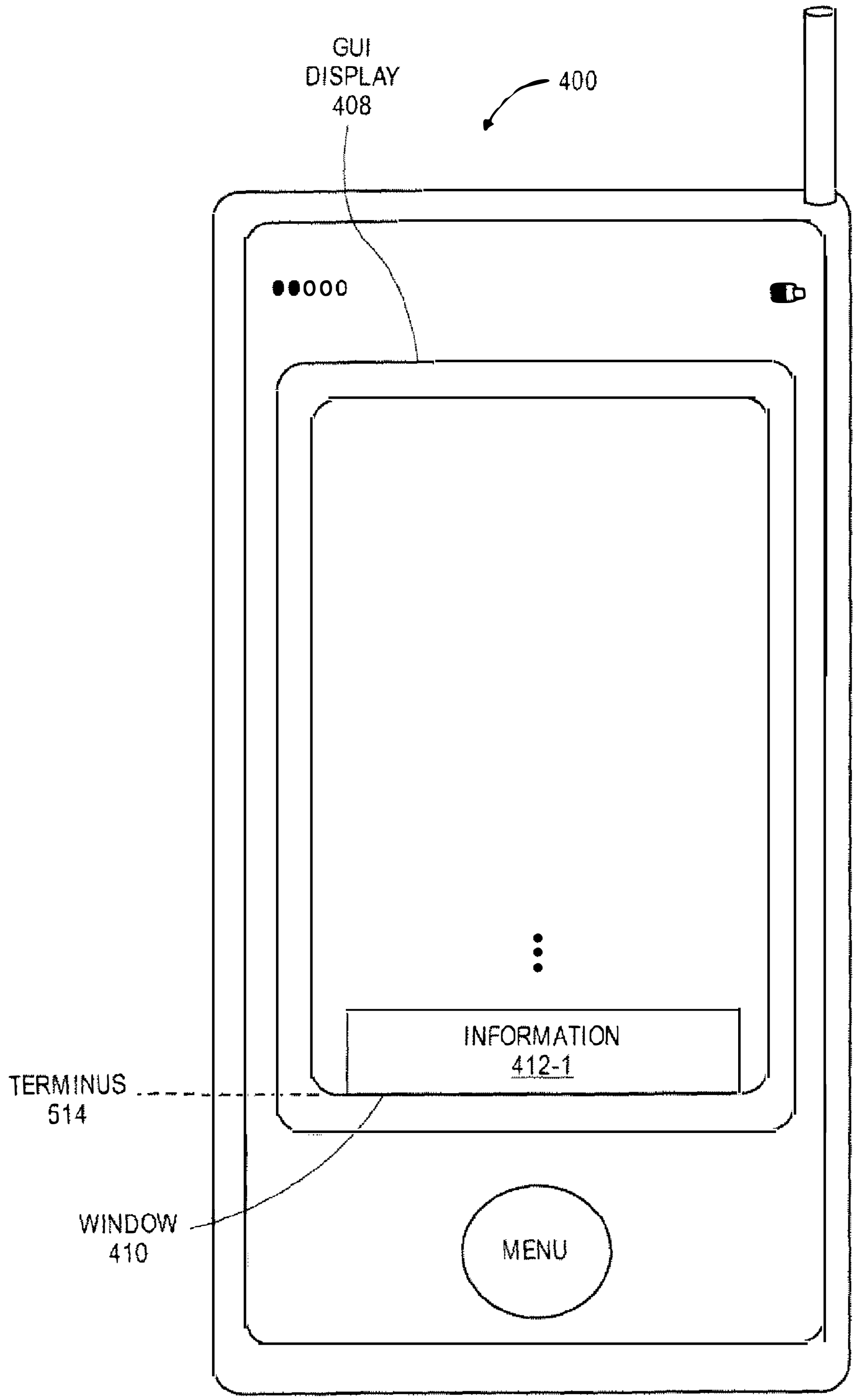

GUI
DISPLAY
408
400
INFORMATION
412-1
TERMINUS
514
WINDOW
410
MENU

FIG. 5B

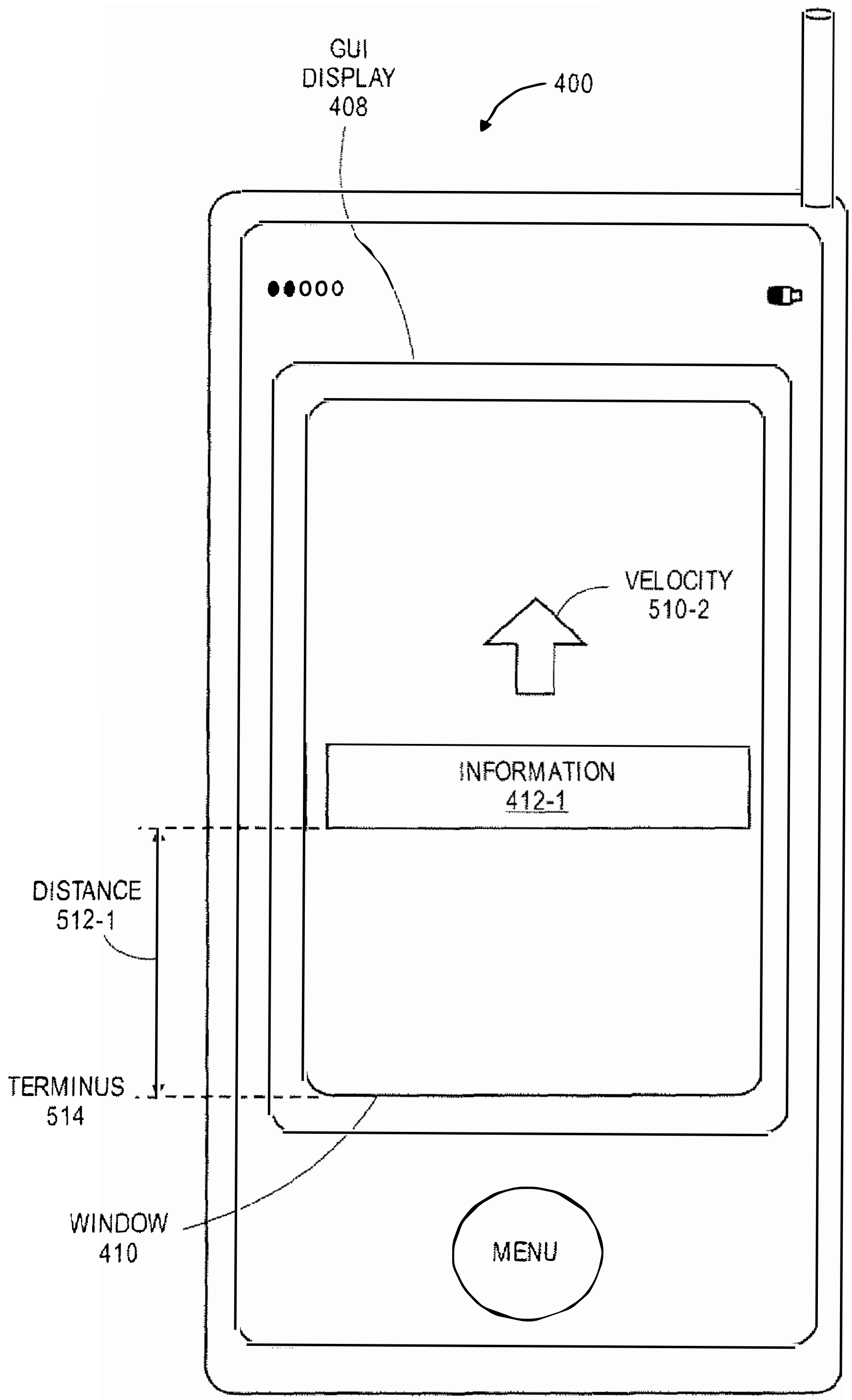

GUI
DISPLAY
408
400
VELOCITY
510-2
INFORMATION
412-1
DISTANCE
512-1
TERMINUS
514
WINDOW
410
MENU

FIG. 5C

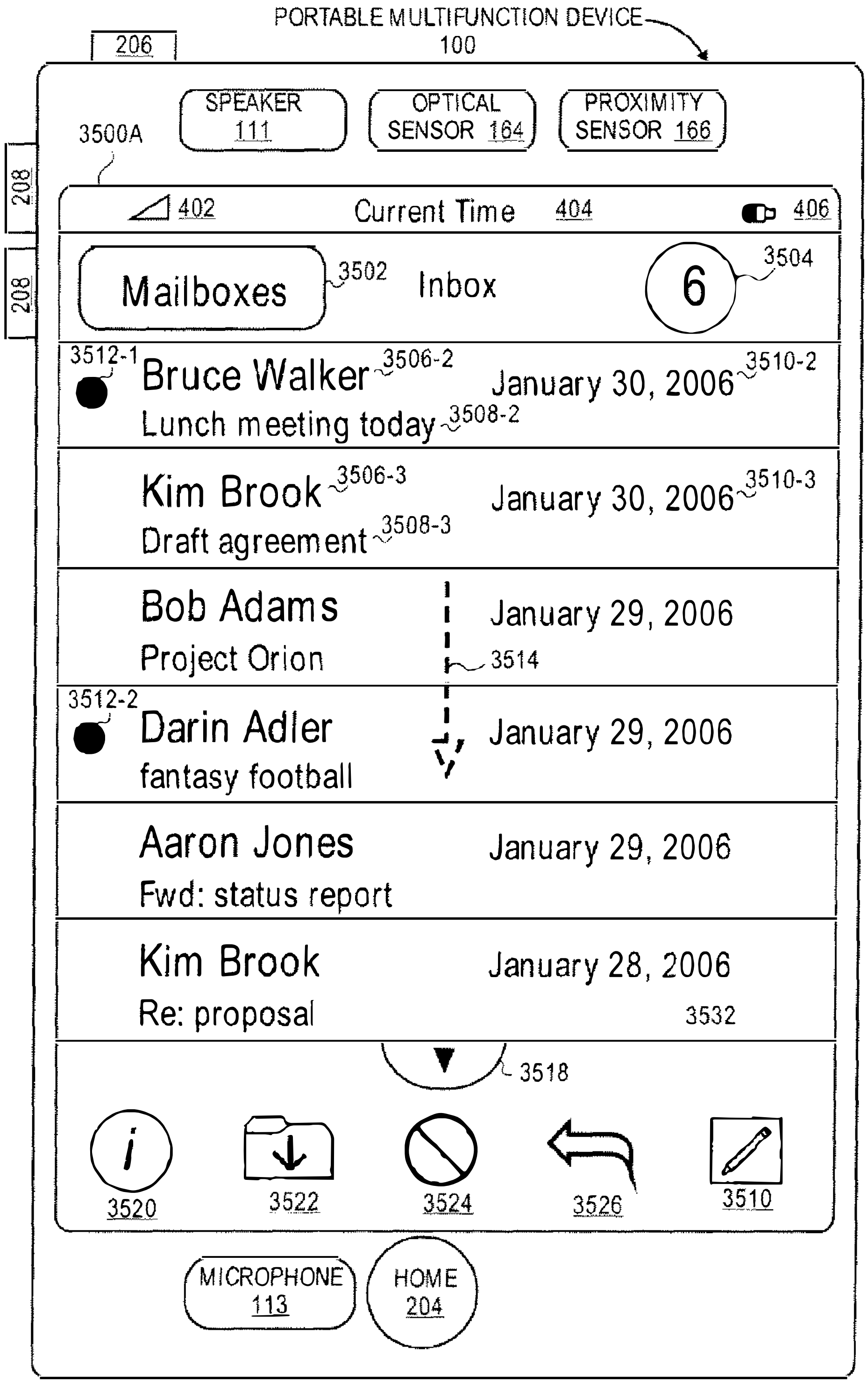
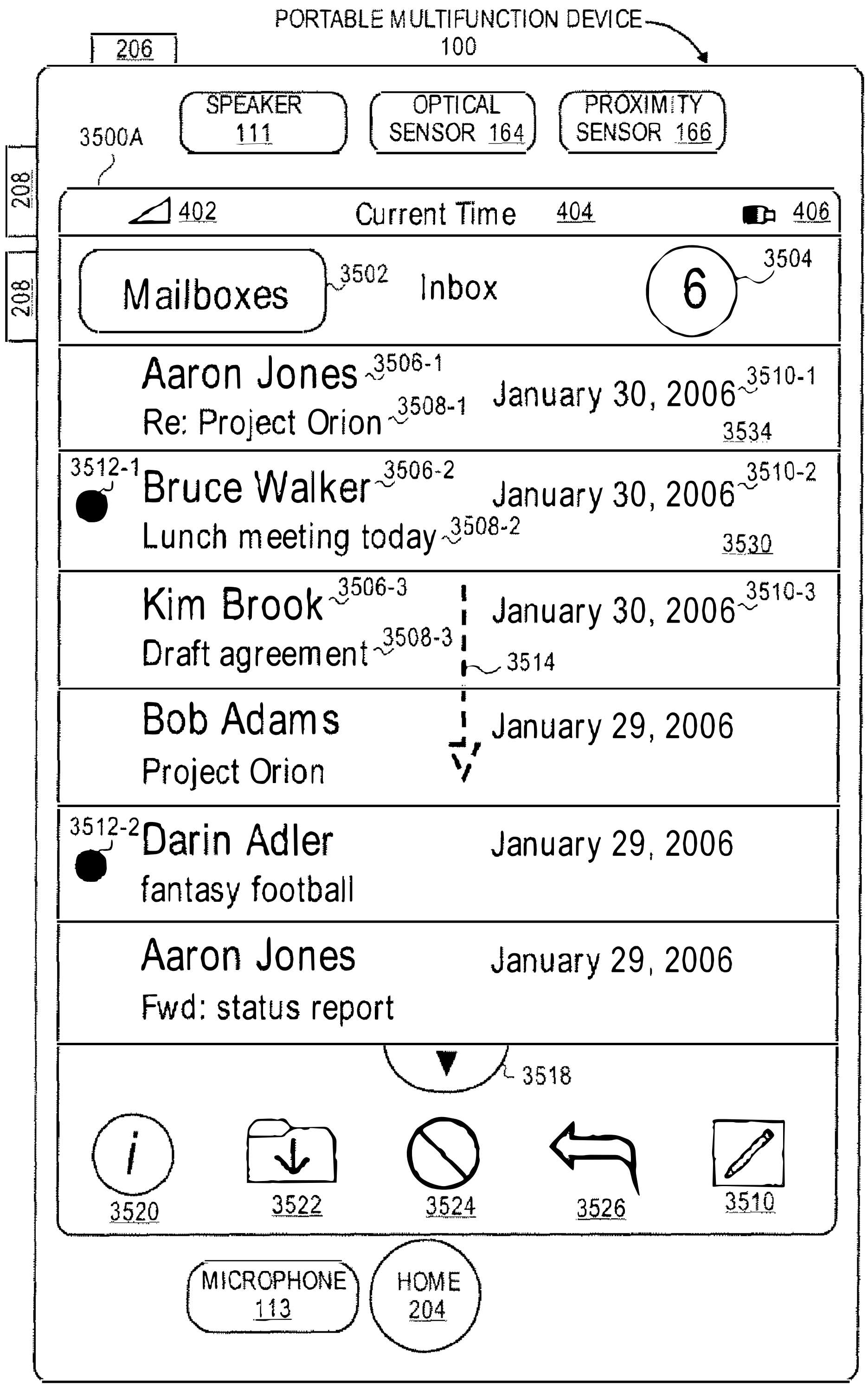
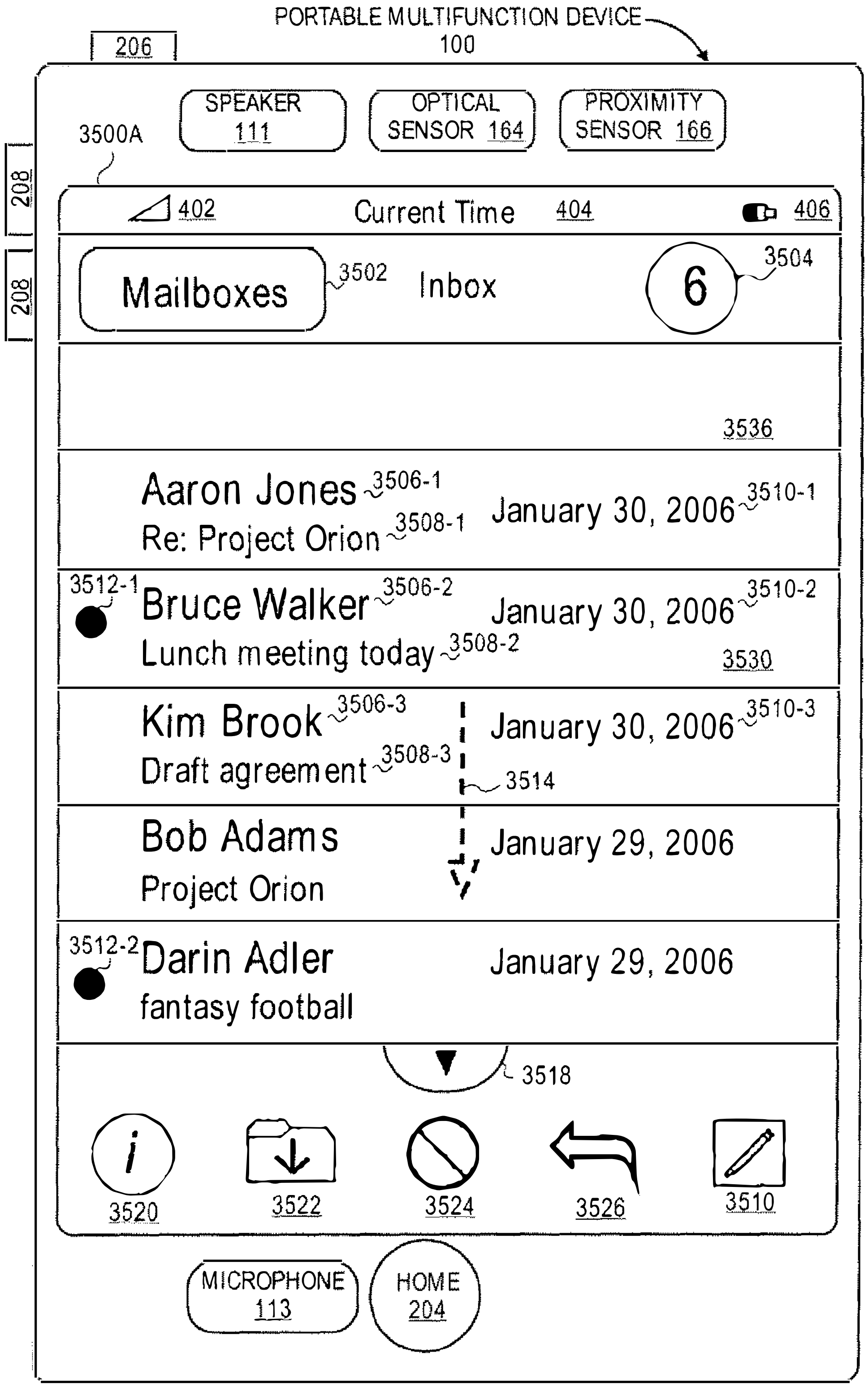

PORTABLE MULTIFUNCTION DEVICE
100
206
3500A
208
208
SPEAKER
111
OPTICAL
SENSOR 164
PROXIMITY
SENSOR 166
402
Current Time 404
406
Mailboxes
3502
Inbox
6
3504
3512-1
Bruce Walker
3506-2
January 30, 2006
3510-2
Lunch meeting today
3508-2
Kim Brook
3506-3
January 30, 2006
3510-3
Draft agreement
3508-3
Bob Adams
January 29, 2006
Project Orion
3514
3512-2
Darin Adler
January 29, 2006
fantasy football
Aaron Jones
January 29, 2006
Fwd: status report
Kim Brook
January 28, 2006
Re: proposal
3532
3518
3520
3522
3524
3526
3510
MICROPHONE
113
HOME
204

FIG. 6A



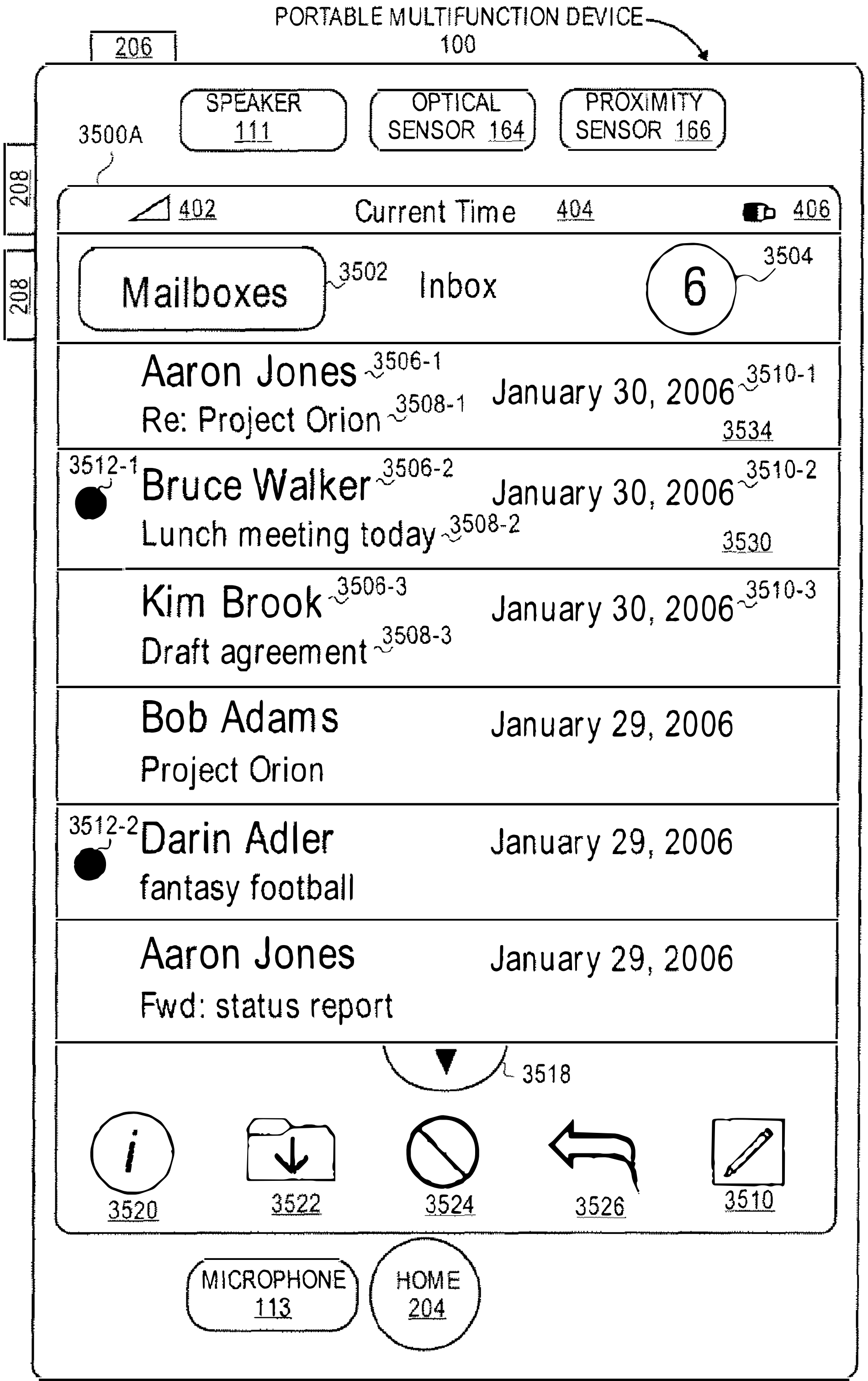
PORTABLE MULTIFUNCTION DEVICE
100
206
208
208
3500A
SPEAKER
111
OPTICAL
SENSOR 164
PROXIMITY
SENSOR 166
402
Current Time
404
406
Mailboxes
3502
Inbox
6
3504
Aaron Jones
3506-1
January 30, 2006
3510-1
Re: Project Orion
3508-1
3534
3512-1
Bruce Walker
3506-2
January 30, 2006
3510-2
Lunch meeting today
3508-2
3530
Kim Brook
3506-3
January 30, 2006
3510-3
Draft agreement
3508-3
3514
Bob Adams
January 29, 2006
Project Orion
3512-2
Darin Adler
January 29, 2006
fantasy football
Aaron Jones
January 29, 2006
Fwd: status report
3518
3520
3522
3524
3526
3510
MICROPHONE
113
HOME
204

FIG. 6B



PORTABLE MULTIFUNCTION DEVICE
100
206
SPEAKER
111
OPTICAL SENSOR 164
PROXIMITY SENSOR 166
3500A
208
208
402
Current Time
404
406
Mailboxes
3502
Inbox
6
3504
3536
Aaron Jones
3506-1
Re: Project Orion
3508-1
January 30, 2006
3510-1
3512-1
Bruce Walker
3506-2
January 30, 2006
3510-2
Lunch meeting today
3508-2
3530
Kim Brook
3506-3
January 30, 2006
3510-3
Draft agreement
3508-3
3514
Bob Adams
January 29, 2006
Project Orion
3512-2
Darin Adler
January 29, 2006
fantasy football
3518
3520
3522
3524
3526
3510
MICROPHONE
113
HOME
204

FIG. 6C



PORTABLE MULTIFUNCTION DEVICE
100
206
208
208
3500A
SPEAKER
111
OPTICAL
SENSOR 164
PROXIMITY
SENSOR 166
402
Current Time
404
406
Mailboxes
3502
Inbox
6
3504
Aaron Jones
3506-1
Re: Project Orion
3508-1
January 30, 2006
3510-1
3534
3512-1
Bruce Walker
3506-2
Lunch meeting today
3508-2
January 30, 2006
3510-2
3530
Kim Brook
3506-3
Draft agreement
3508-3
January 30, 2006
3510-3
Bob Adams
Project Orion
January 29, 2006
3512-2
Darin Adler
fantasy football
January 29, 2006
Aaron Jones
Fwd: status report
January 29, 2006
3518
3520
3522
3524
3526
3510
MICROPHONE
113
HOME
204

FIG. 6D

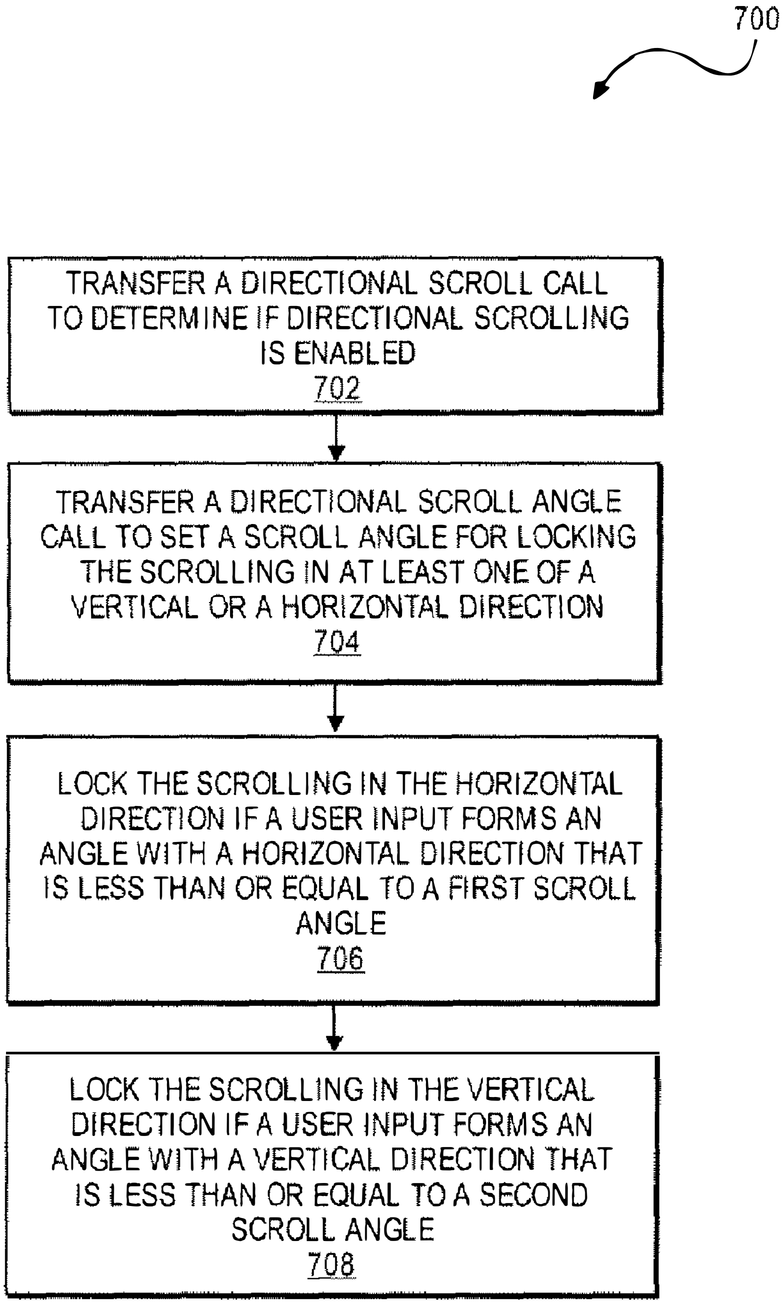

700
TRANSFER A DIRECTIONAL SCROLL CALL TO DETERMINE IF DIRECTIONAL SCROLLING IS ENABLED
702
TRANSFER A DIRECTIONAL SCROLL ANGLE CALL TO SET A SCROLL ANGLE FOR LOCKING THE SCROLLING IN AT LEAST ONE OF A VERTICAL OR A HORIZONTAL DIRECTION
704
LOCK THE SCROLLING IN THE HORIZONTAL DIRECTION IF A USER INPUT FORMS AN ANGLE WITH A HORIZONTAL DIRECTION THAT IS LESS THAN OR EQUAL TO A FIRST SCROLL ANGLE
706
LOCK THE SCROLLING IN THE VERTICAL DIRECTION IF A USER INPUT FORMS AN ANGLE WITH A VERTICAL DIRECTION THAT IS LESS THAN OR EQUAL TO A SECOND SCROLL ANGLE
708

FIG. 7

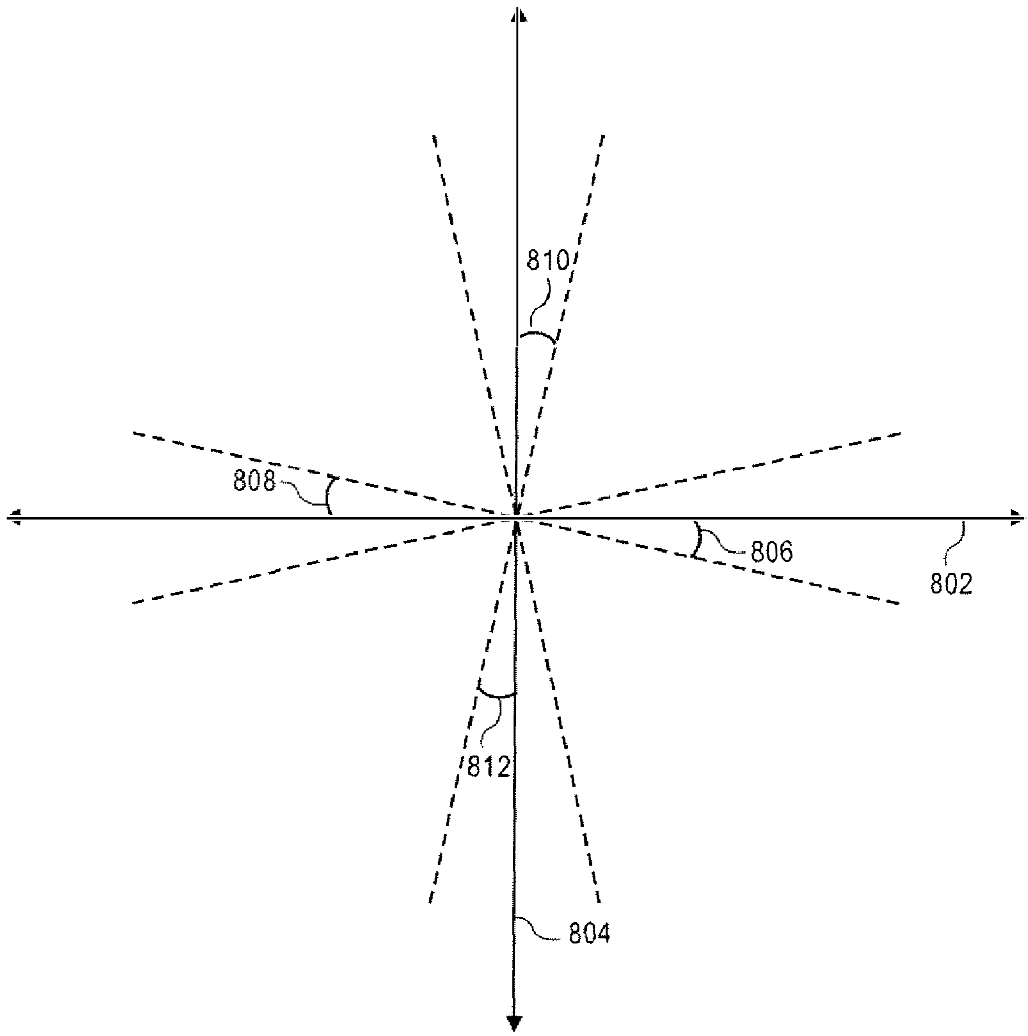


FIG. 8

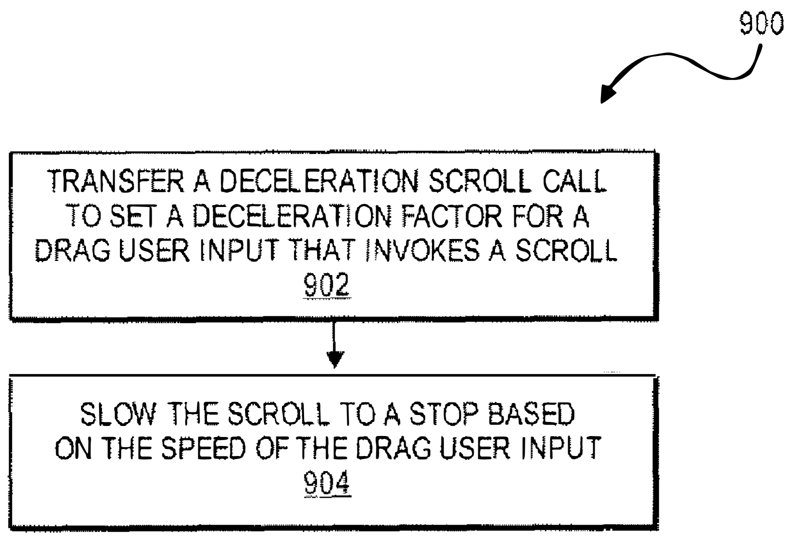

900
TRANSFER A DECELERATION SCROLL CALL TO SET A DECELERATION FACTOR FOR A DRAG USER INPUT THAT INVOKES A SCROLL
902
SLOW THE SCROLL TO A STOP BASED ON THE SPEED OF THE DRAG USER INPUT
904

FIG. 9

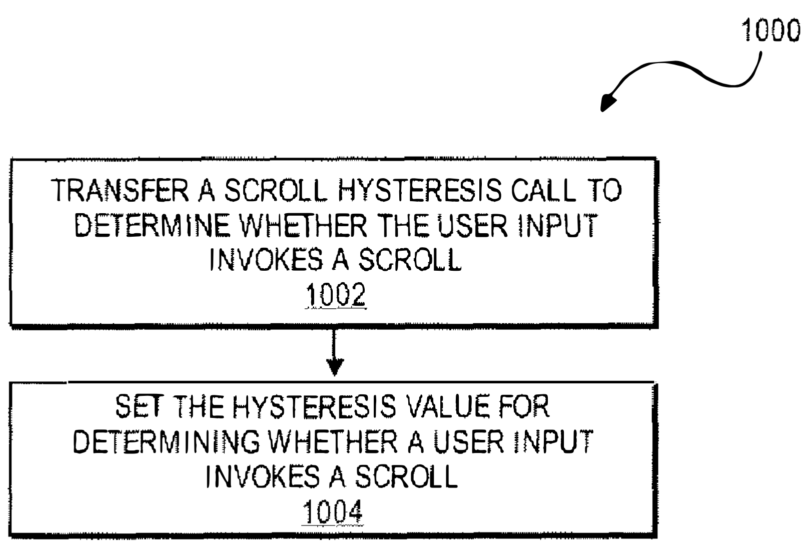

1000
TRANSFER A SCROLL HYSTERESIS CALL TO DETERMINE WHETHER THE USER INPUT INVOKES A SCROLL
1002
SET THE HYSTERESIS VALUE FOR DETERMINING WHETHER A USER INPUT INVOKES A SCROLL
1004

FIG. 10

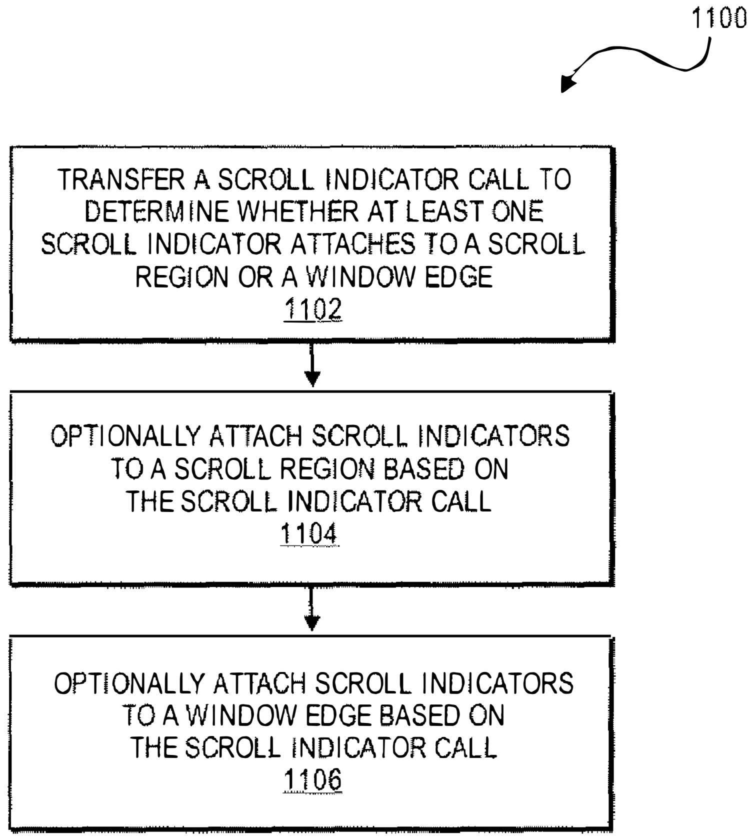

1100
TRANSFER A SCROLL INDICATOR CALL TO DETERMINE WHETHER AT LEAST ONE SCROLL INDICATOR ATTACHES TO A SCROLL REGION OR A WINDOW EDGE
1102
OPTIONALLY ATTACH SCROLL INDICATORS TO A SCROLL REGION BASED ON THE SCROLL INDICATOR CALL
1104
OPTIONALLY ATTACH SCROLL INDICATORS TO A WINDOW EDGE BASED ON THE SCROLL INDICATOR CALL
1106

FIG. 11

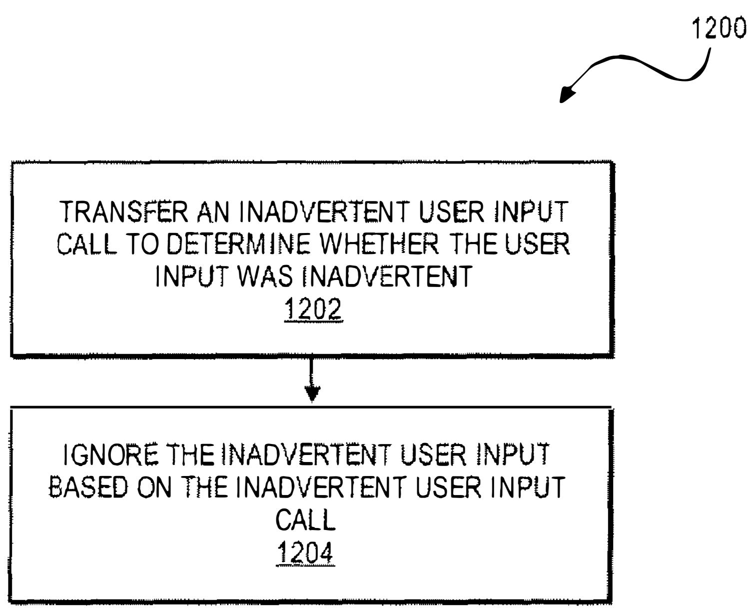

1200
TRANSFER AN INADVERTENT USER INPUT CALL TO DETERMINE WHETHER THE USER INPUT WAS INADVERTENT
1202
IGNORE THE INADVERTENT USER INPUT BASED ON THE INADVERTENT USER INPUT CALL
1204

FIG. 12

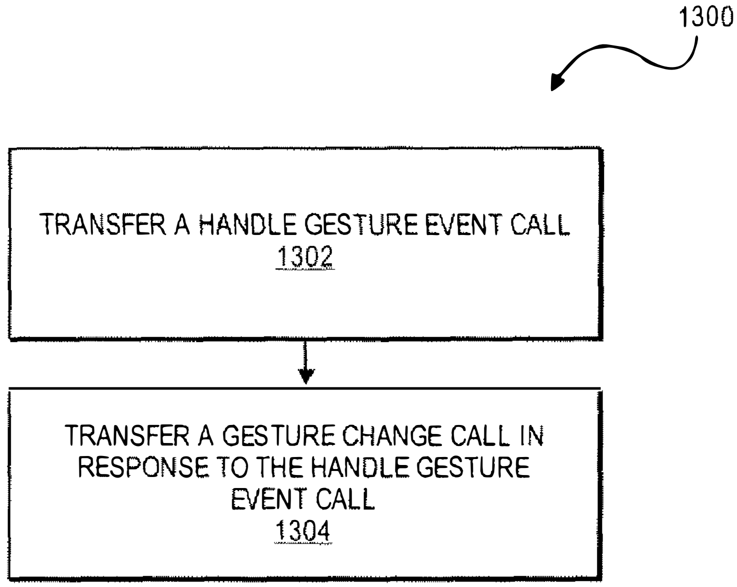

1300
TRANSFER A HANDLE GESTURE EVENT CALL
1302
TRANSFER A GESTURE CHANGE CALL IN RESPONSE TO THE HANDLE GESTURE EVENT CALL
1304

FIG. 13

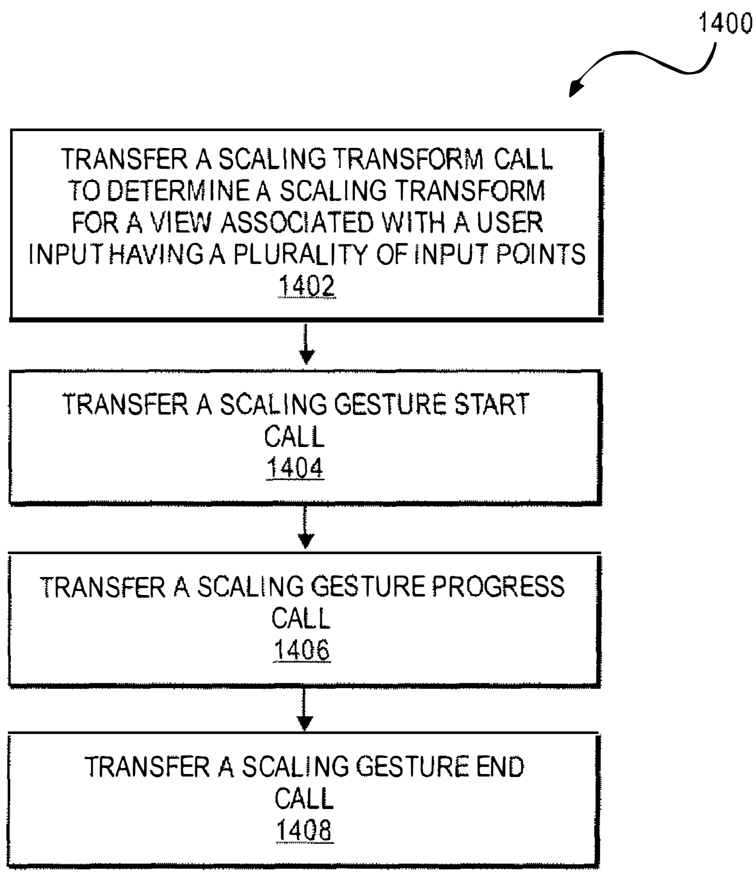

1400
TRANSFER A SCALING TRANSFORM CALL TO DETERMINE A SCALING TRANSFORM FOR A VIEW ASSOCIATED WITH A USER INPUT HAVING A PLURALITY OF INPUT POINTS
1402
TRANSFER A SCALING GESTURE START CALL
1404
TRANSFER A SCALING GESTURE PROGRESS CALL
1406
TRANSFER A SCALING GESTURE END CALL
1408

FIG. 14

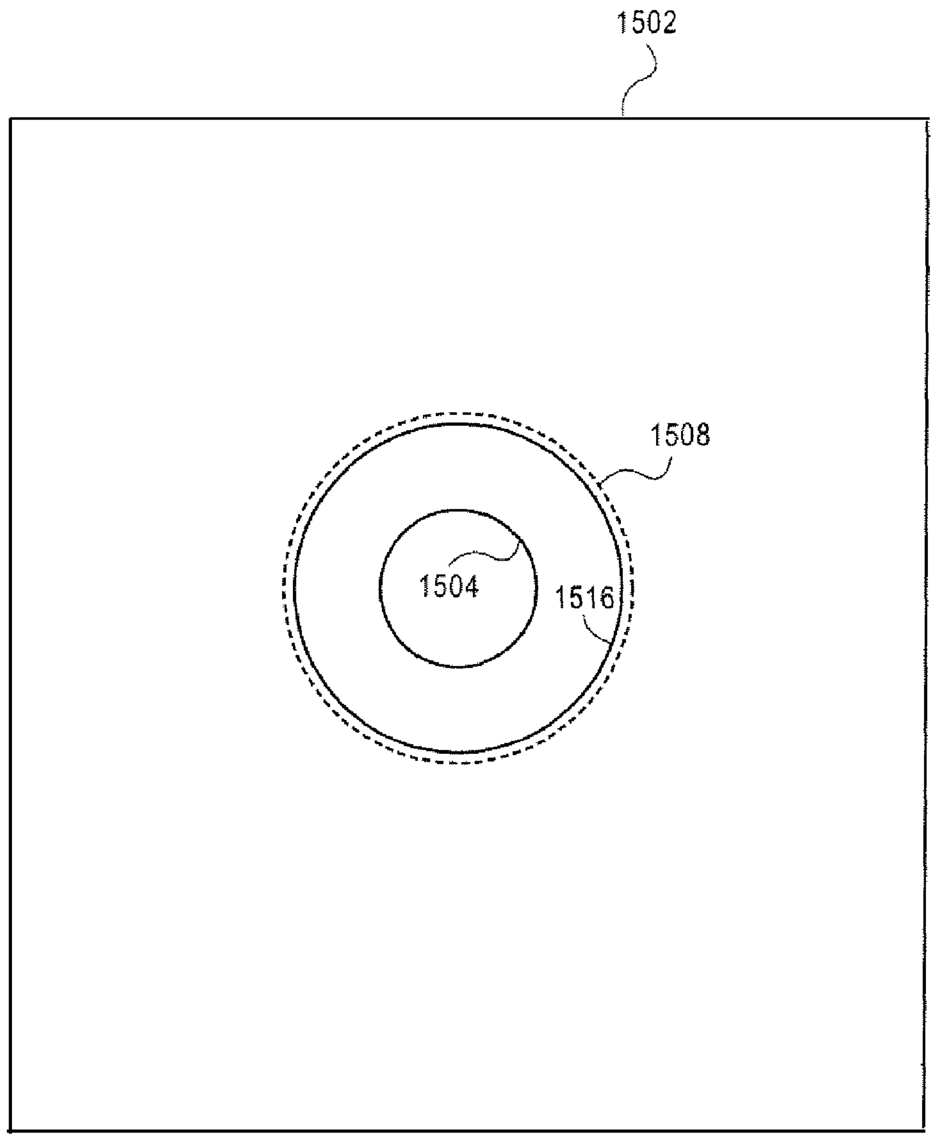


FIG. 15

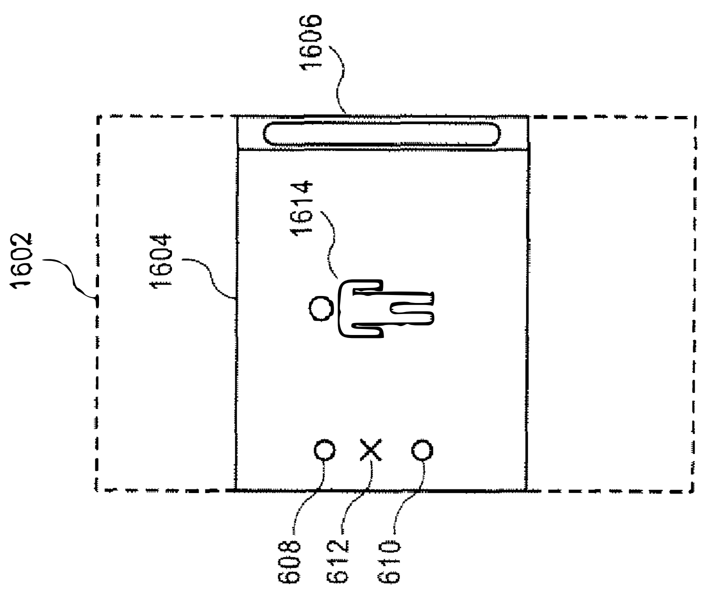


FIG. 16A

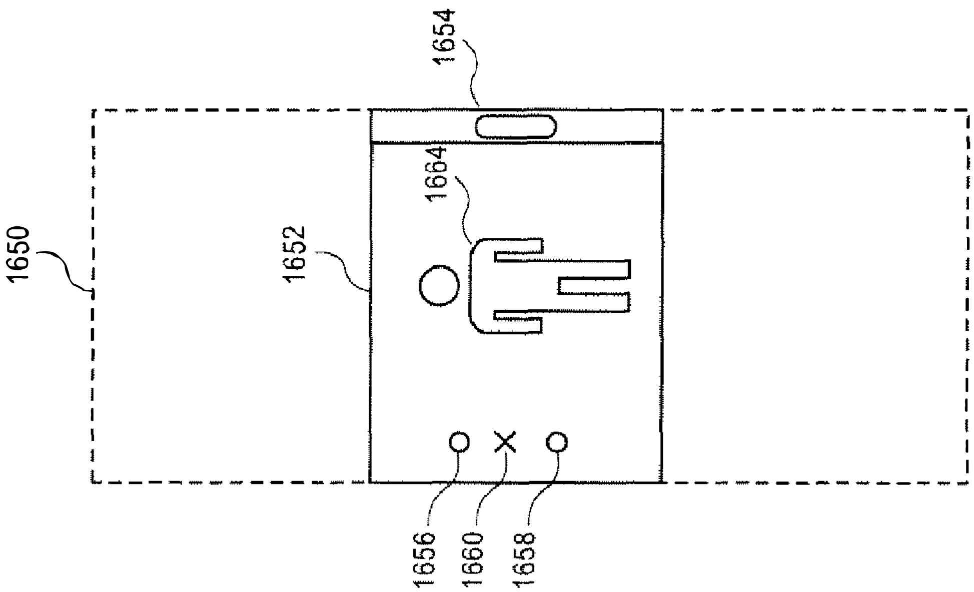


FIG. 16B

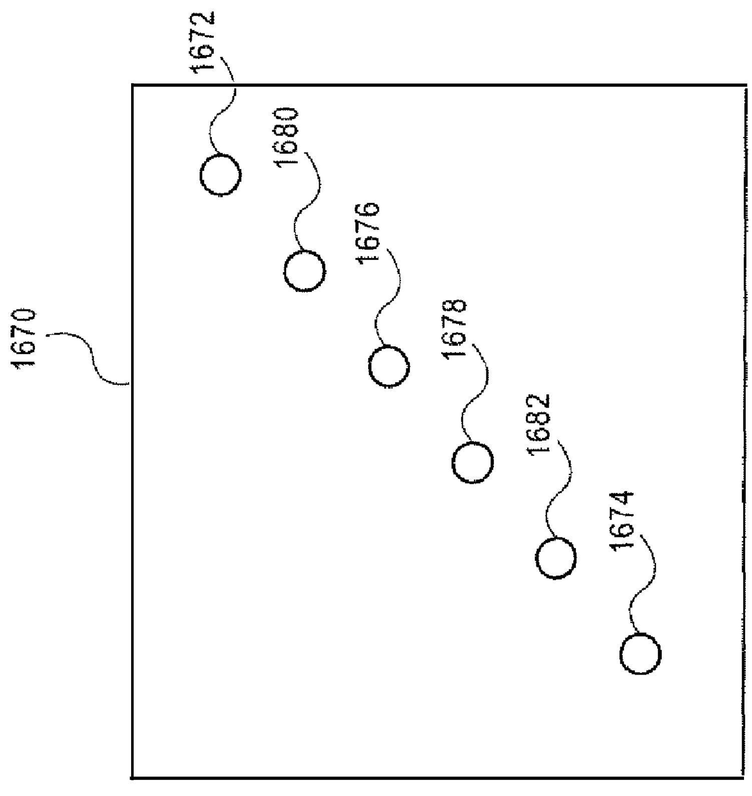


FIG. 16C

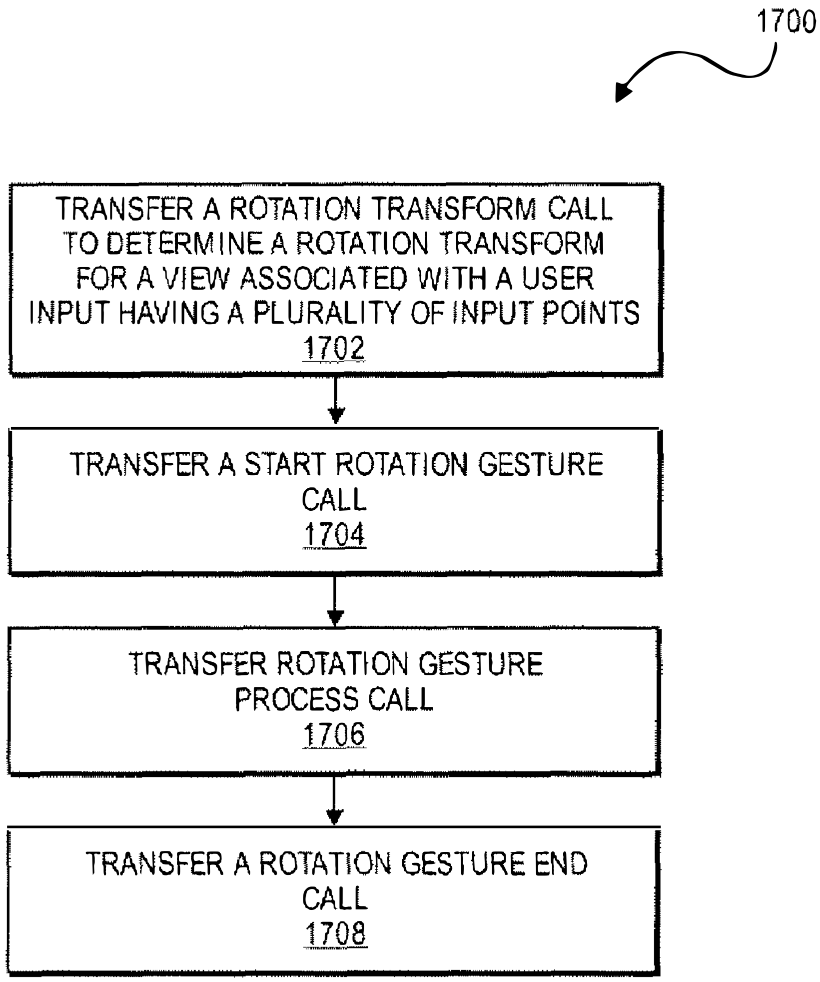

1700
TRANSFER A ROTATION TRANSFORM CALL TO DETERMINE A ROTATION TRANSFORM FOR A VIEW ASSOCIATED WITH A USER INPUT HAVING A PLURALITY OF INPUT POINTS
1702
TRANSFER A START ROTATION GESTURE CALL
1704
TRANSFER ROTATION GESTURE PROCESS CALL
1706
TRANSFER A ROTATION GESTURE END CALL
1708

FIG. 17

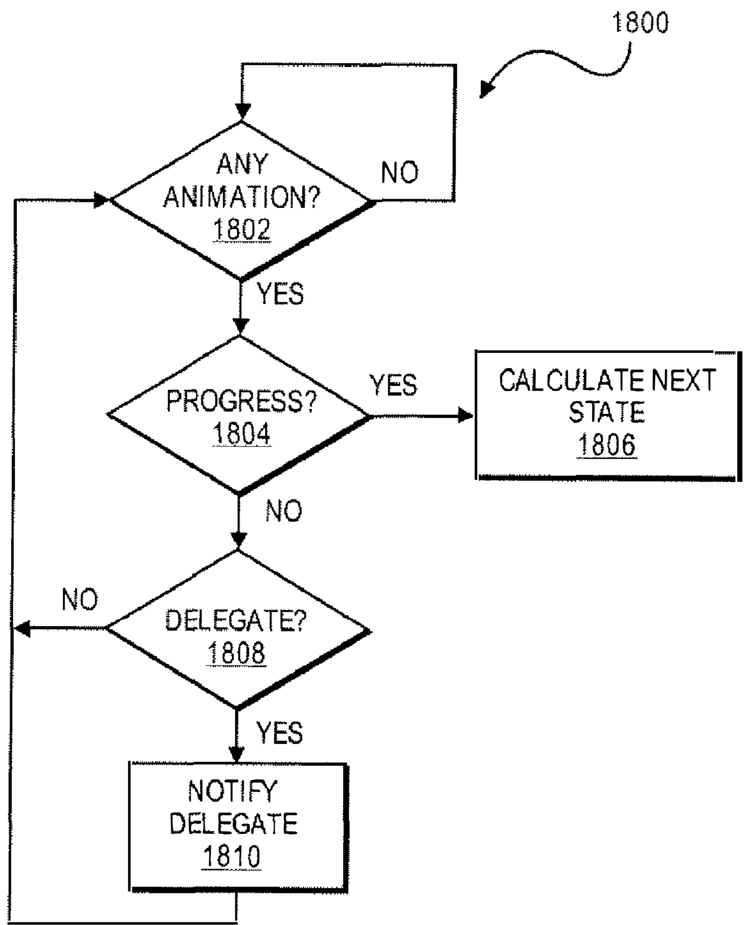

1800
ANY ANIMATION? 1802
NO
YES
PROGRESS? 1804
YES
CALCULATE NEXT STATE 1806
NO
DELEGATE? 1808
NO
YES
NOTIFY DELEGATE 1810

FIG. 18

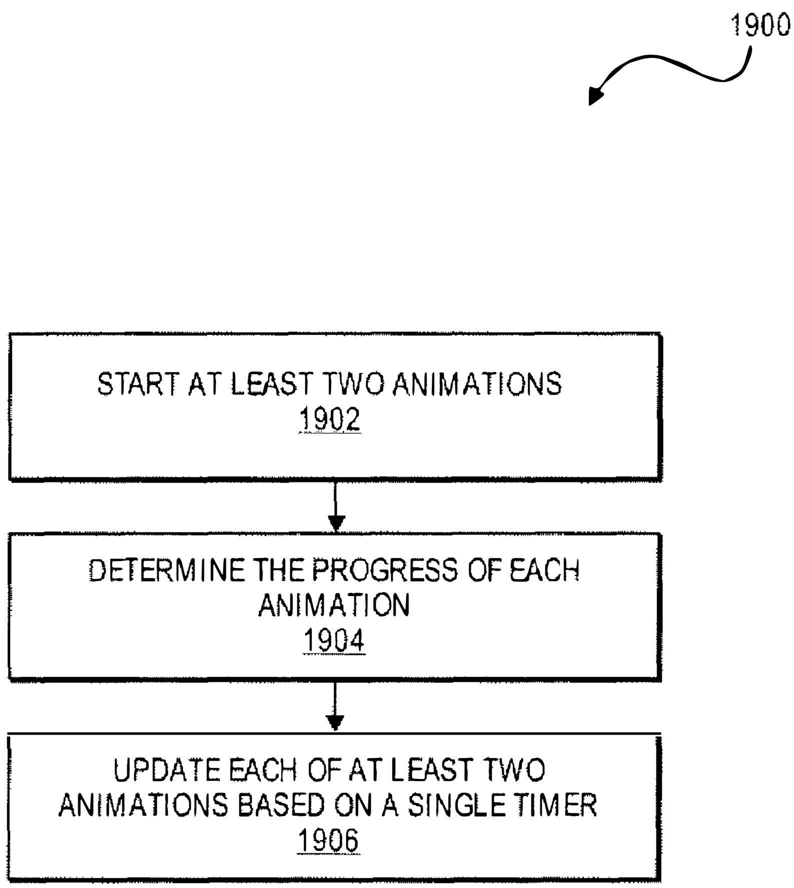

1900
START AT LEAST TWO ANIMATIONS
1902
DETERMINE THE PROGRESS OF EACH ANIMATION
1904
UPDATE EACH OF AT LEAST TWO ANIMATIONS BASED ON A SINGLE TIMER
1906

FIG. 19

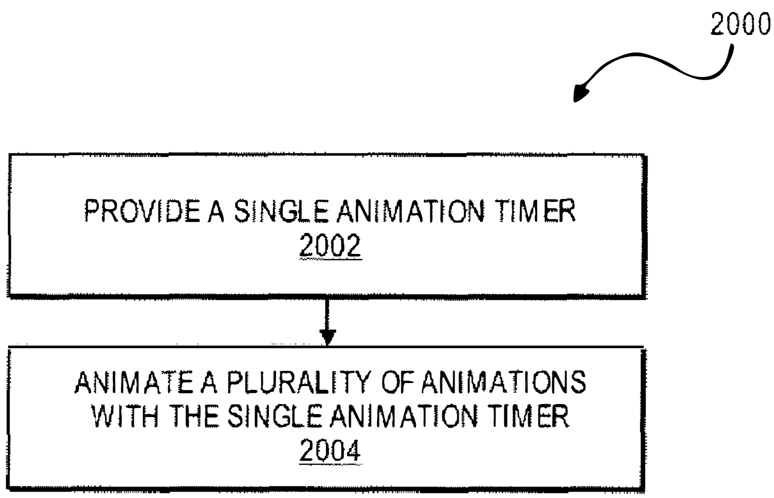

2000
PROVIDE A SINGLE ANIMATION TIMER
2002
ANIMATE A PLURALITY OF ANIMATIONS WITH THE SINGLE ANIMATION TIMER
2004

FIG. 20

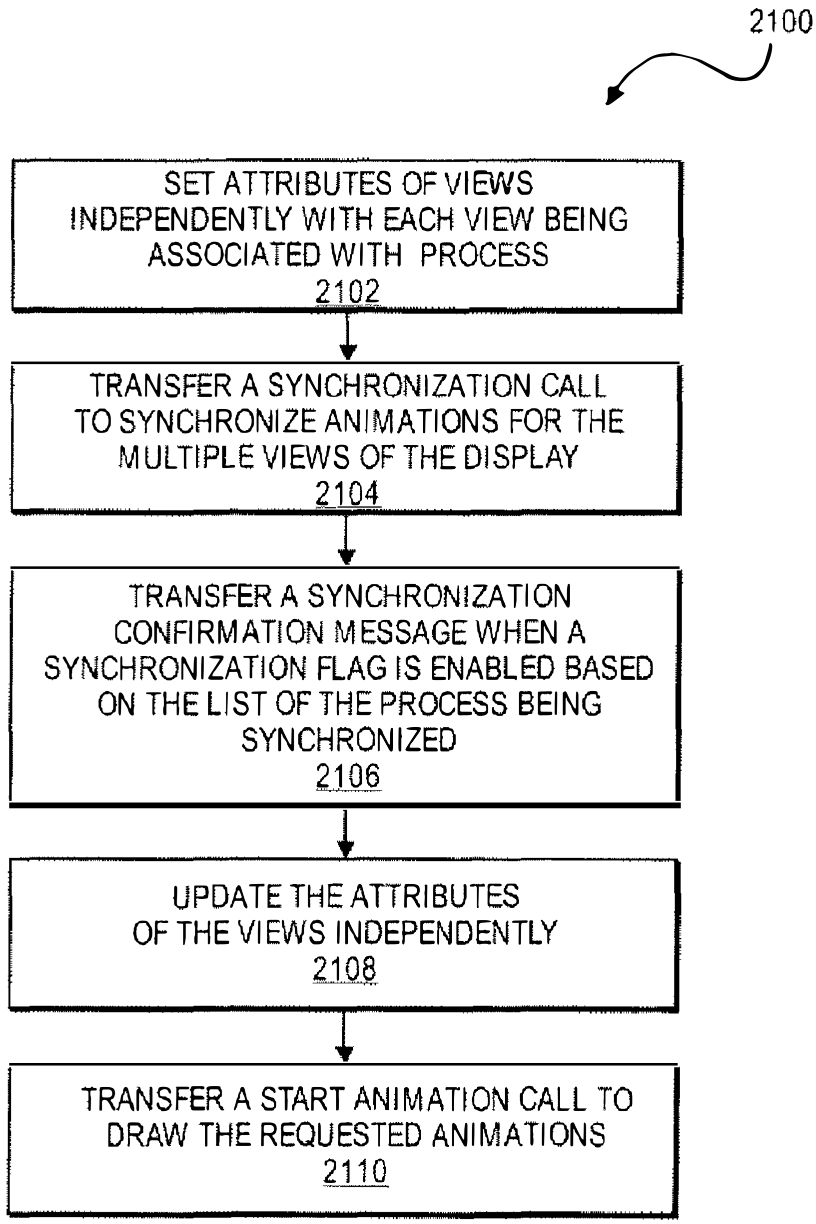

2100
SET ATTRIBUTES OF VIEWS INDEPENDENTLY WITH EACH VIEW BEING ASSOCIATED WITH PROCESS
2102
TRANSFER A SYNCHRONIZATION CALL TO SYNCHRONIZE ANIMATIONS FOR THE MULTIPLE VIEWS OF THE DISPLAY
2104
TRANSFER A SYNCHRONIZATION CONFIRMATION MESSAGE WHEN A SYNCHRONIZATION FLAG IS ENABLED BASED ON THE LIST OF THE PROCESS BEING SYNCHRONIZED
2106
UPDATE THE ATTRIBUTES OF THE VIEWS INDEPENDENTLY
2108
TRANSFER A START ANIMATION CALL TO DRAW THE REQUESTED ANIMATIONS
2110

FIG. 21

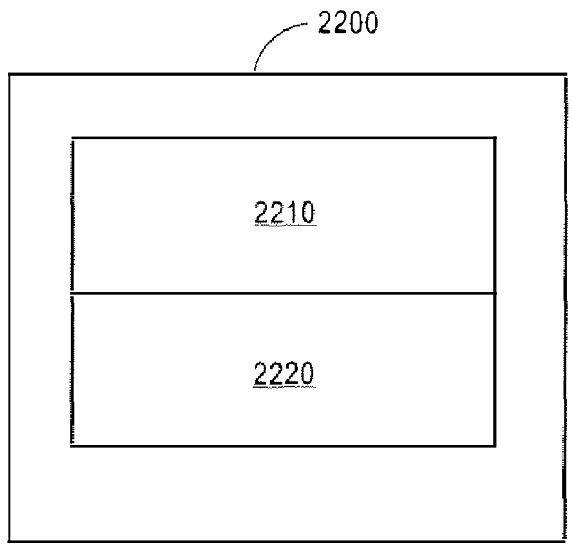


FIG. 22A

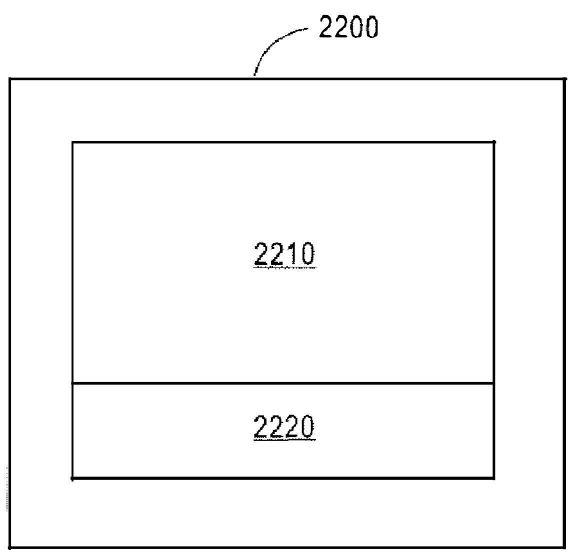


FIG. 22B

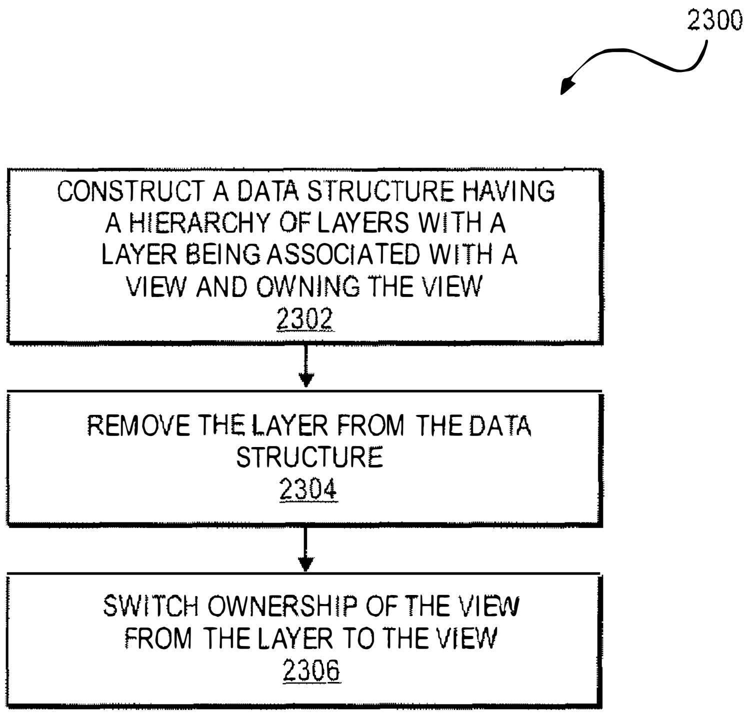

2300
CONSTRUCT A DATA STRUCTURE HAVING A HIERARCHY OF LAYERS WITH A LAYER BEING ASSOCIATED WITH A VIEW AND OWNING THE VIEW
2302
REMOVE THE LAYER FROM THE DATA STRUCTURE
2304
SWITCH OWNERSHIP OF THE VIEW FROM THE LAYER TO THE VIEW
2306

FIG. 23

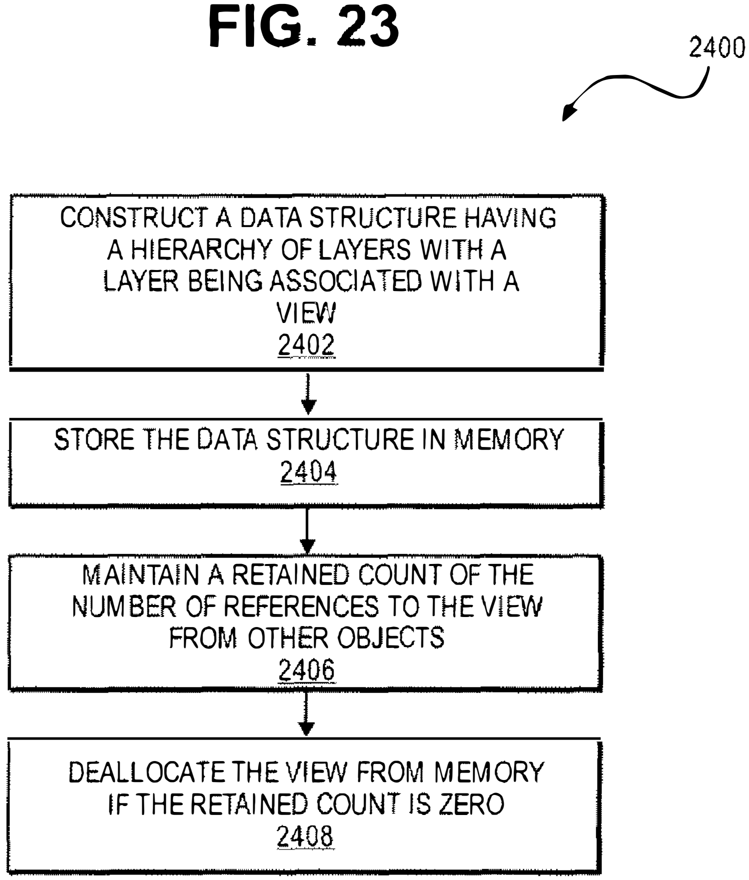

2400
CONSTRUCT A DATA STRUCTURE HAVING A HIERARCHY OF LAYERS WITH A LAYER BEING ASSOCIATED WITH A VIEW
2402
STORE THE DATA STRUCTURE IN MEMORY
2404
MAINTAIN A RETAINED COUNT OF THE NUMBER OF REFERENCES TO THE VIEW FROM OTHER OBJECTS
2406
DEALLOCATE THE VIEW FROM MEMORY IF THE RETAINED COUNT IS ZERO
2408

FIG. 24

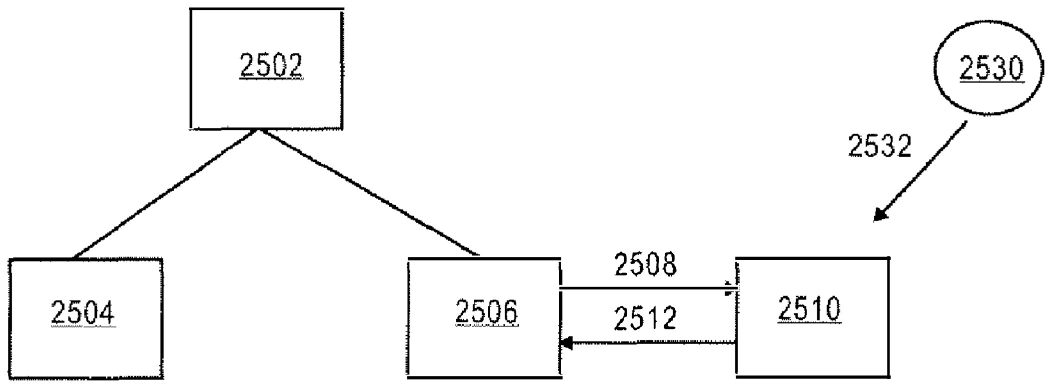


FIG. 25A

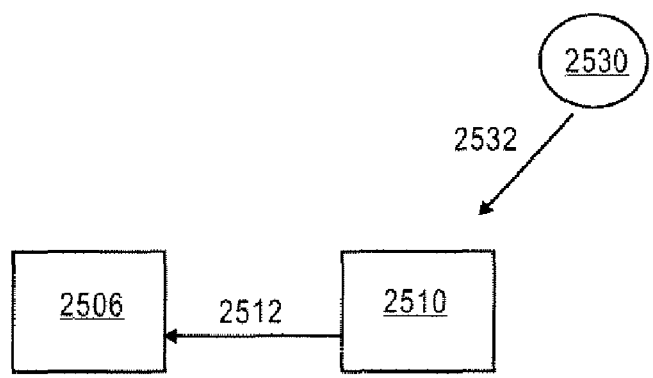


FIG. 25B

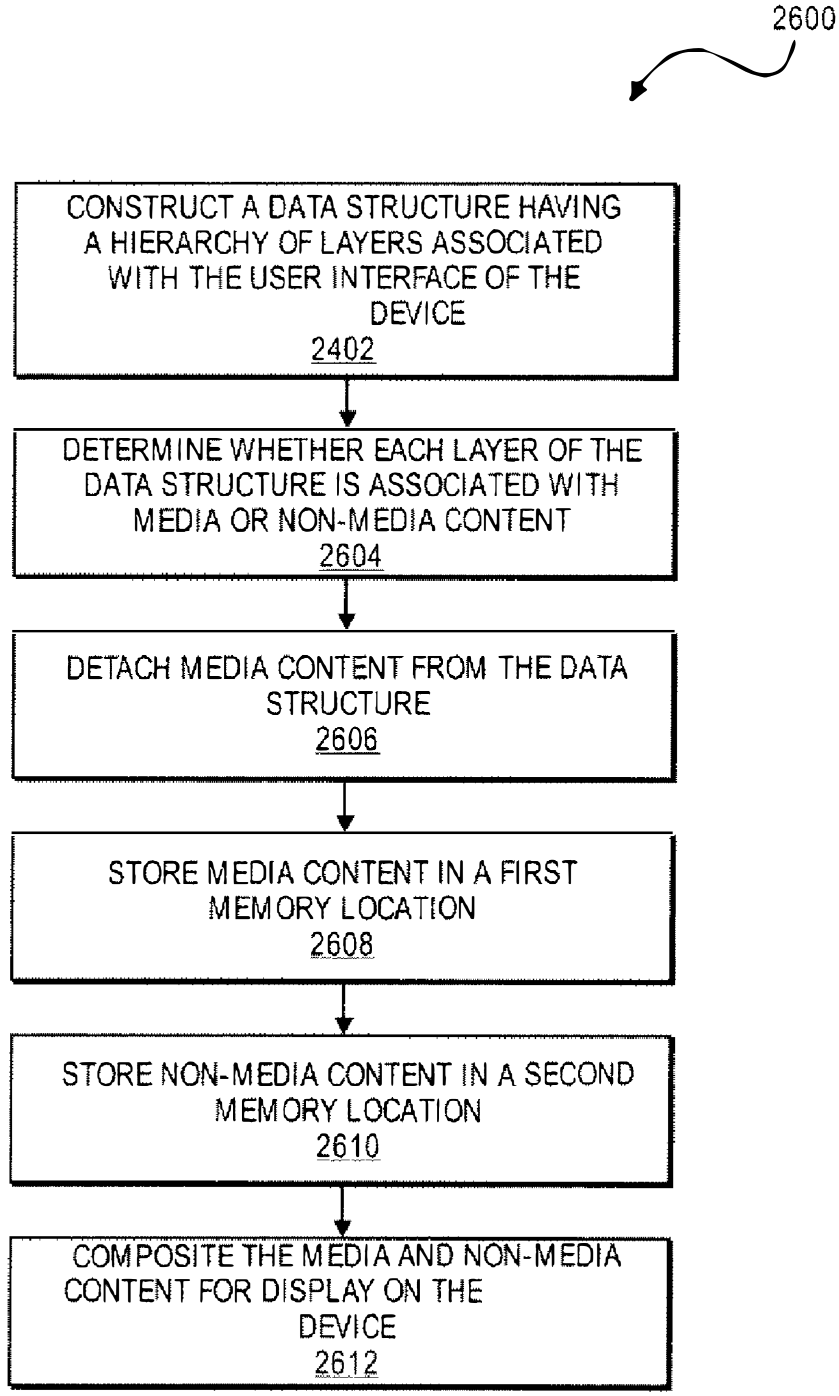

2600
CONSTRUCT A DATA STRUCTURE HAVING A HIERARCHY OF LAYERS ASSOCIATED WITH THE USER INTERFACE OF THE DEVICE
2402
DETERMINE WHETHER EACH LAYER OF THE DATA STRUCTURE IS ASSOCIATED WITH MEDIA OR NON-MEDIA CONTENT
2604
DETACH MEDIA CONTENT FROM THE DATA STRUCTURE
2606
STORE MEDIA CONTENT IN A FIRST MEMORY LOCATION
2608
STORE NON-MEDIA CONTENT IN A SECOND MEMORY LOCATION
2610
COMPOSITE THE MEDIA AND NON-MEDIA CONTENT FOR DISPLAY ON THE DEVICE
2612

FIG. 26

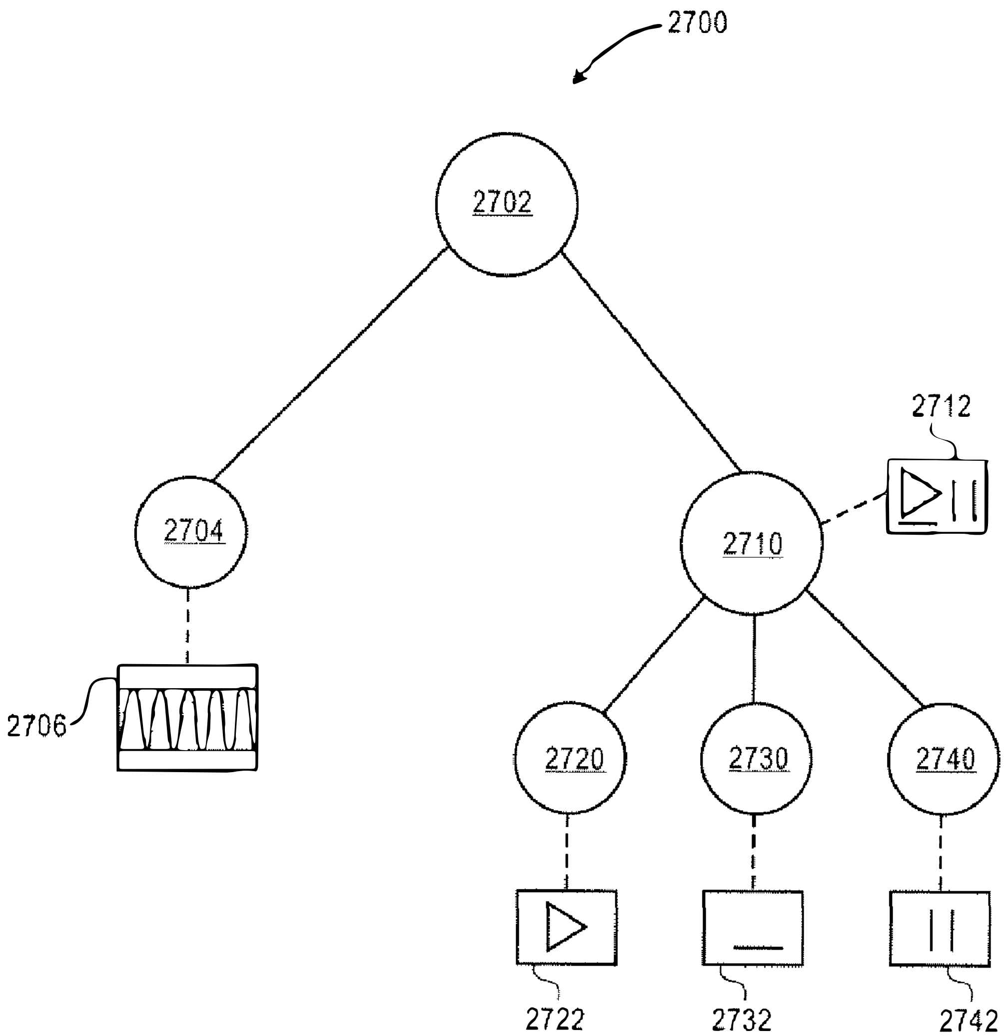


FIG. 27

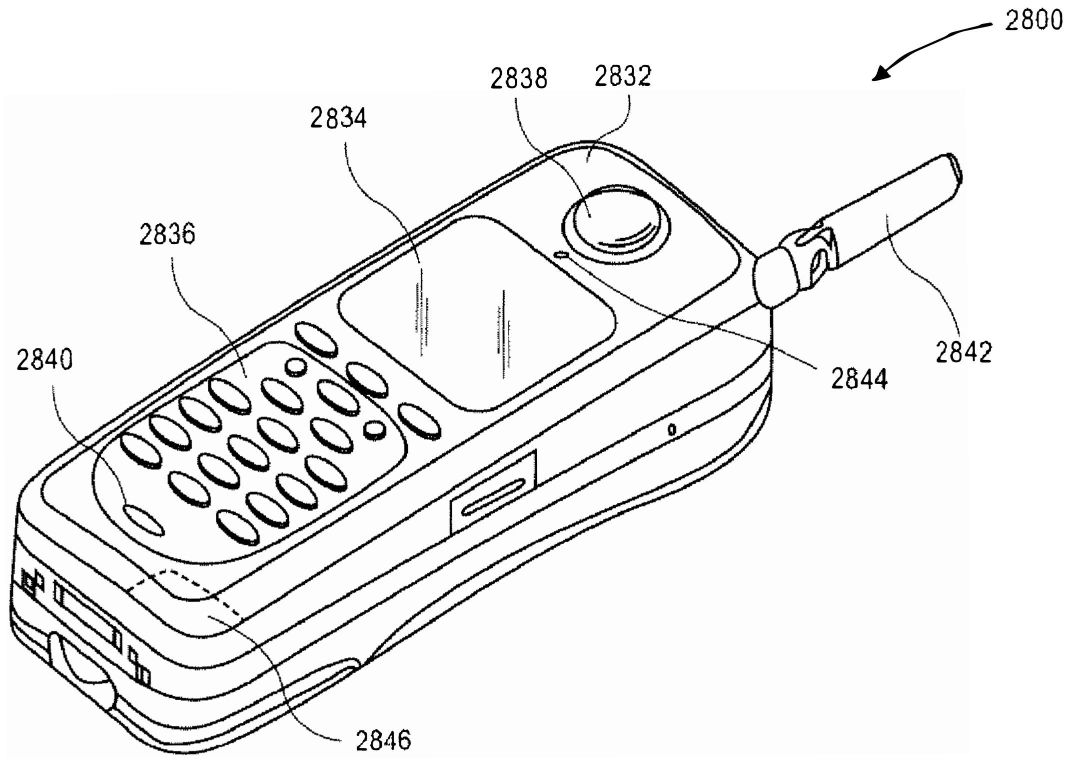


FIG. 28

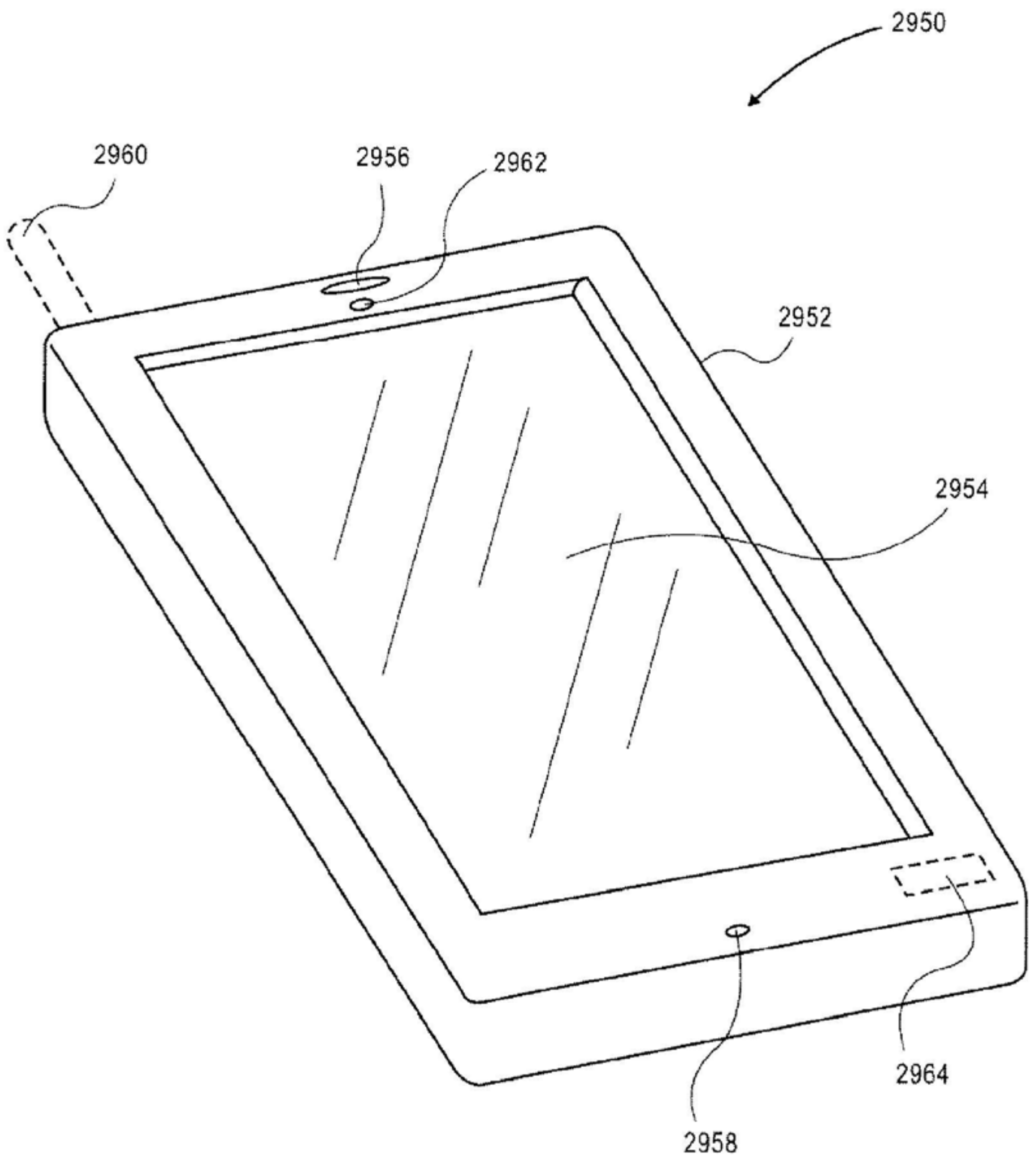


FIG. 29

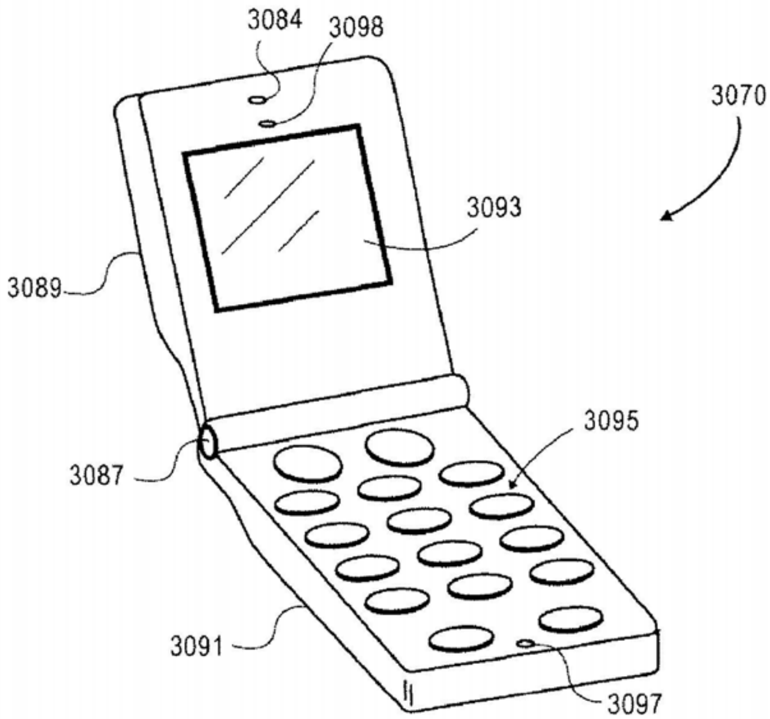


FIG. 30A

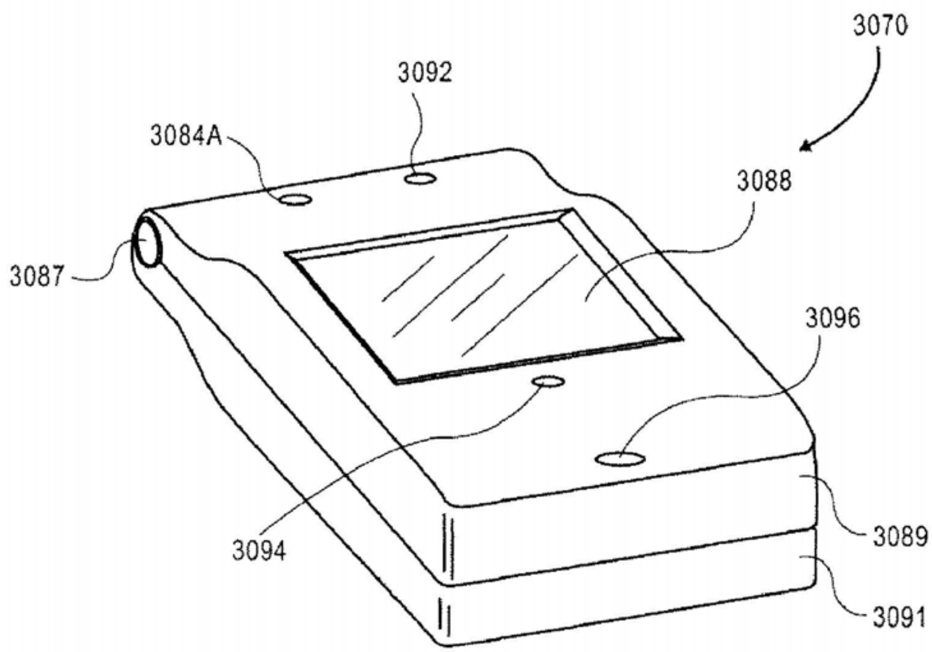


FIG. 30B

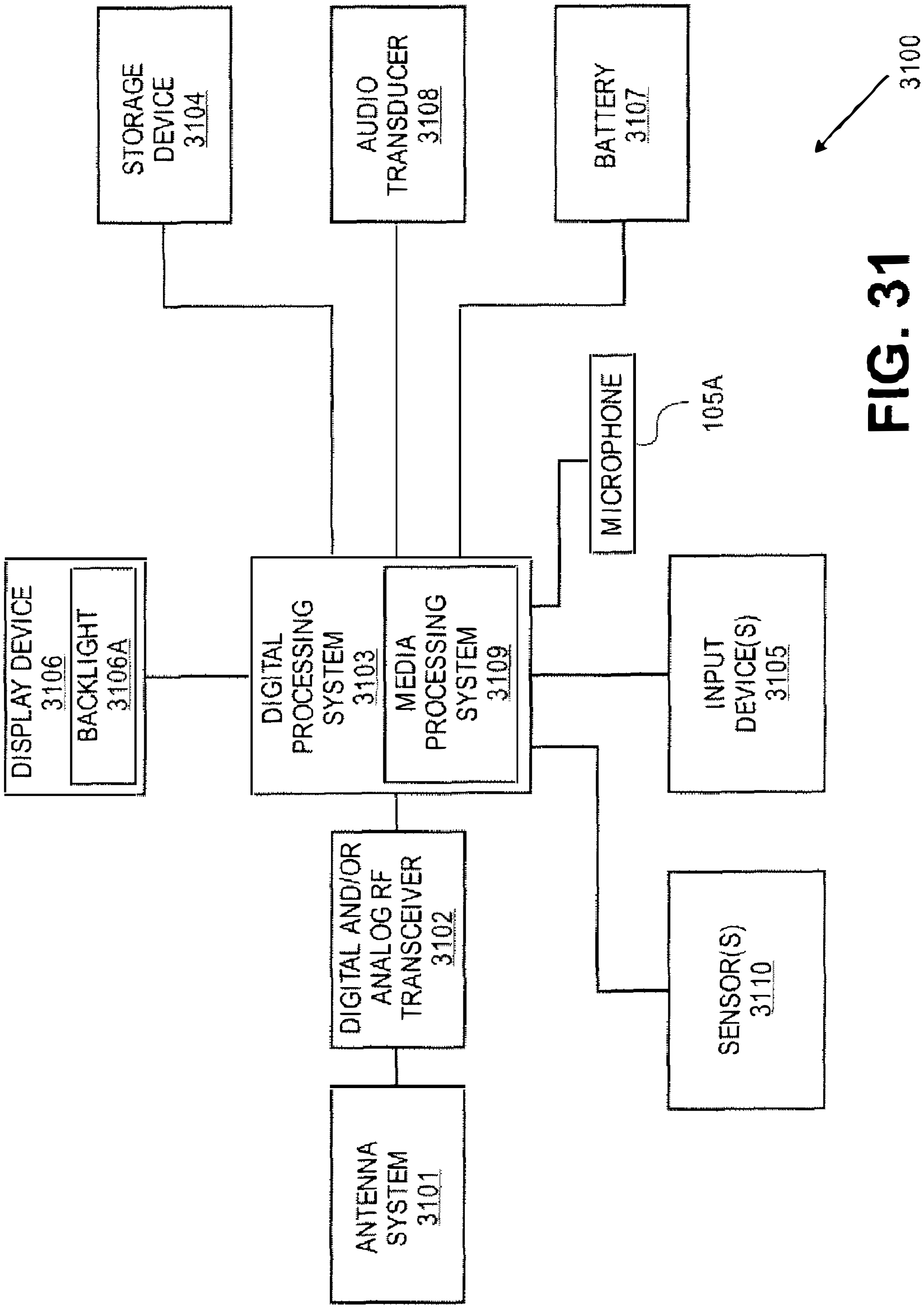

STORAGE DEVICE 3104
AUDIO TRANSDUCER 3108
BATTERY 3107
3100
MICROPHONE
105A
DISPLAY DEVICE 3106
BACKLIGHT 3106A
DIGITAL PROCESSING SYSTEM 3103
MEDIA PROCESSING SYSTEM 3109
INPUT DEVICE(S) 3105
DIGITAL AND/OR ANALOG RF TRANSCEIVER 3102
SENSOR(S) 3110
ANTENNA SYSTEM 3101

FIG. 31

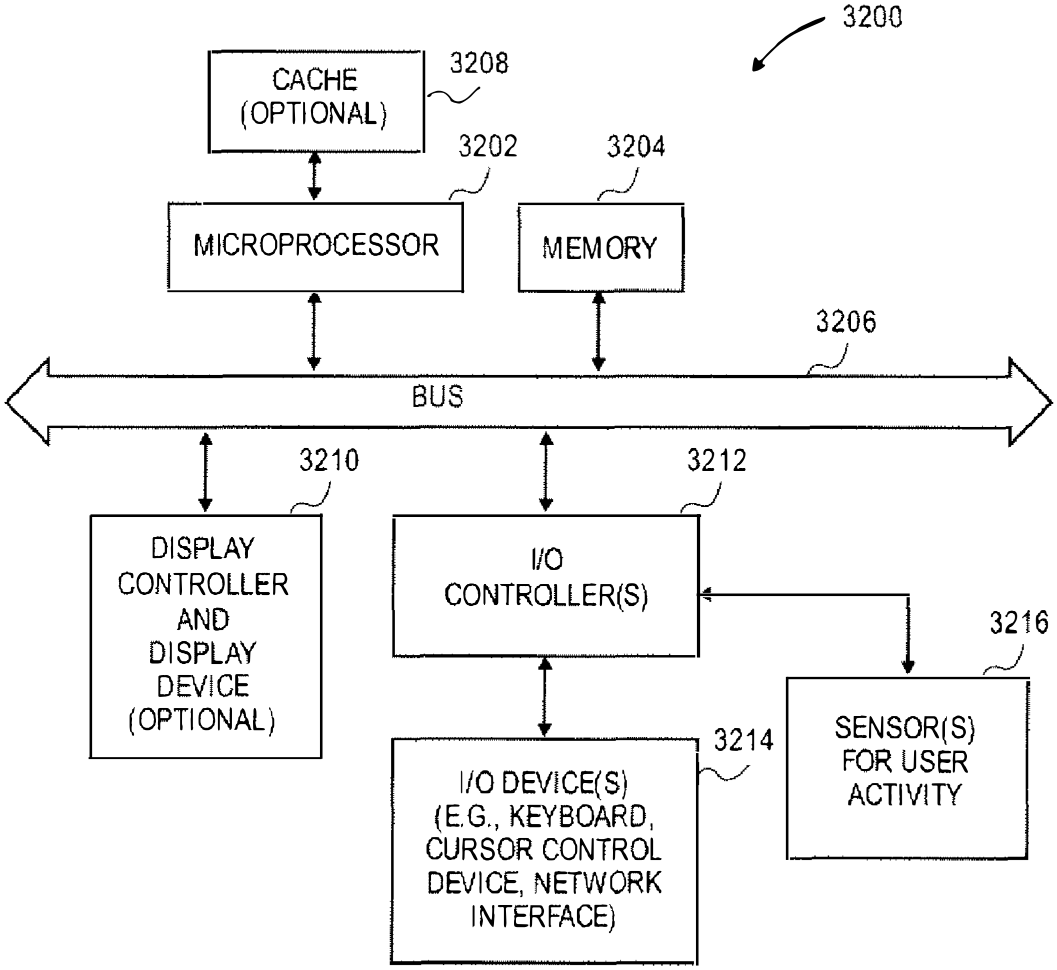

3200
CACHE (OPTIONAL)
3208
3202
3204
MICROPROCESSOR
MEMORY
3206
BUS
3210
3212
DISPLAY CONTROLLER AND DISPLAY DEVICE (OPTIONAL)
I/O CONTROLLER(S)
3216
SENSOR(S) FOR USER ACTIVITY
3214
I/O DEVICE(S) (E.G., KEYBOARD, CURSOR CONTROL DEVICE, NETWORK INTERFACE)

FIG. 32

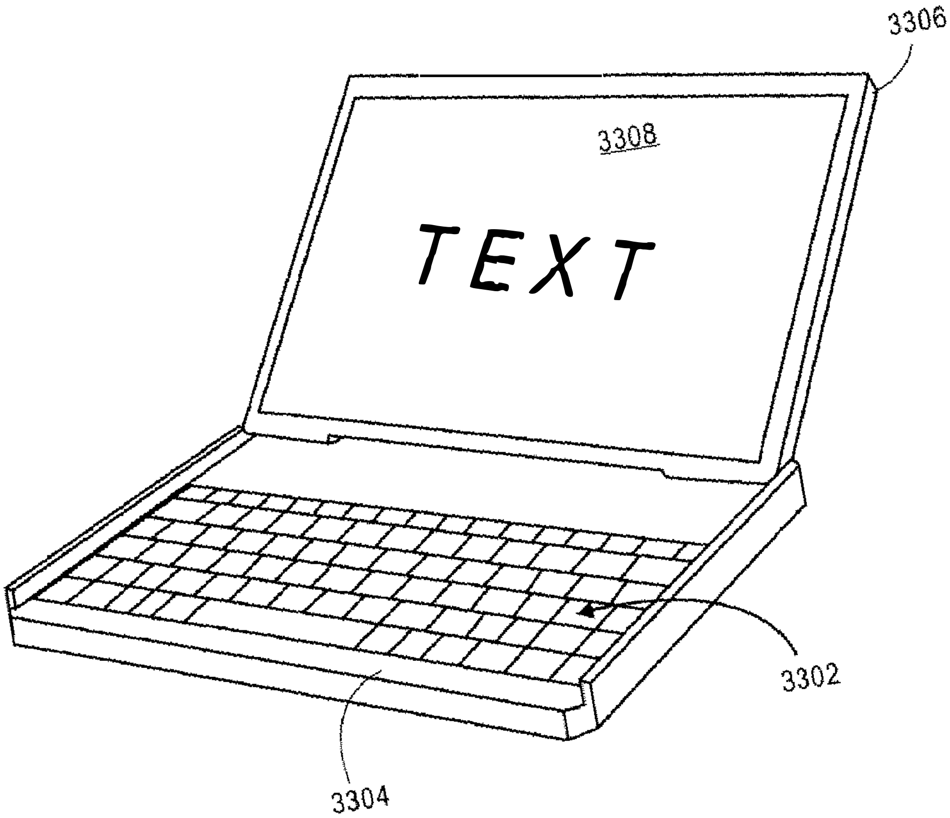


FIG. 33A

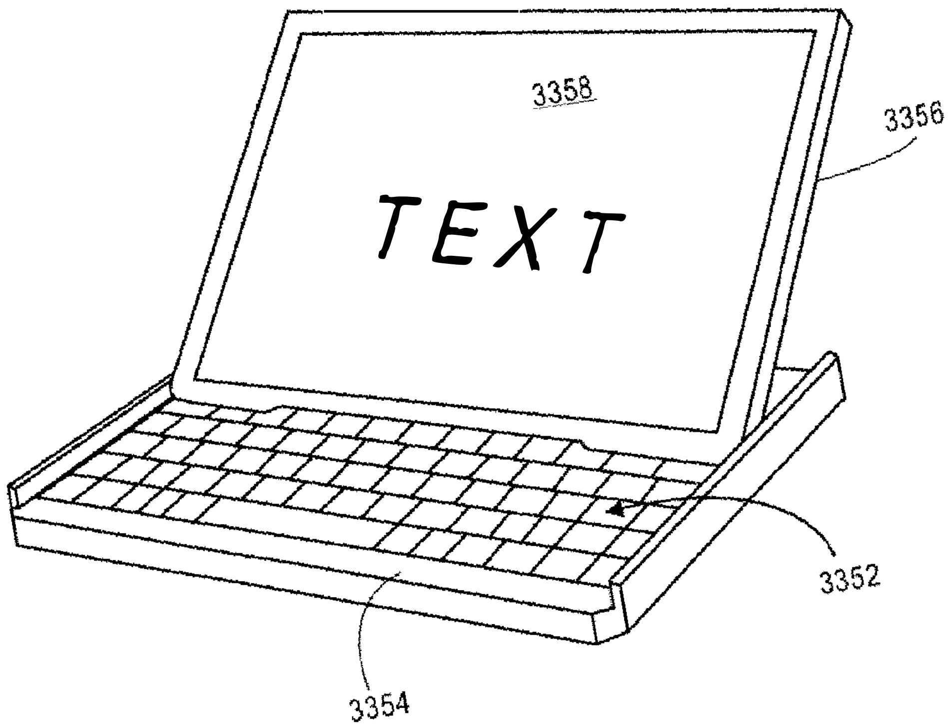


FIG. 33B

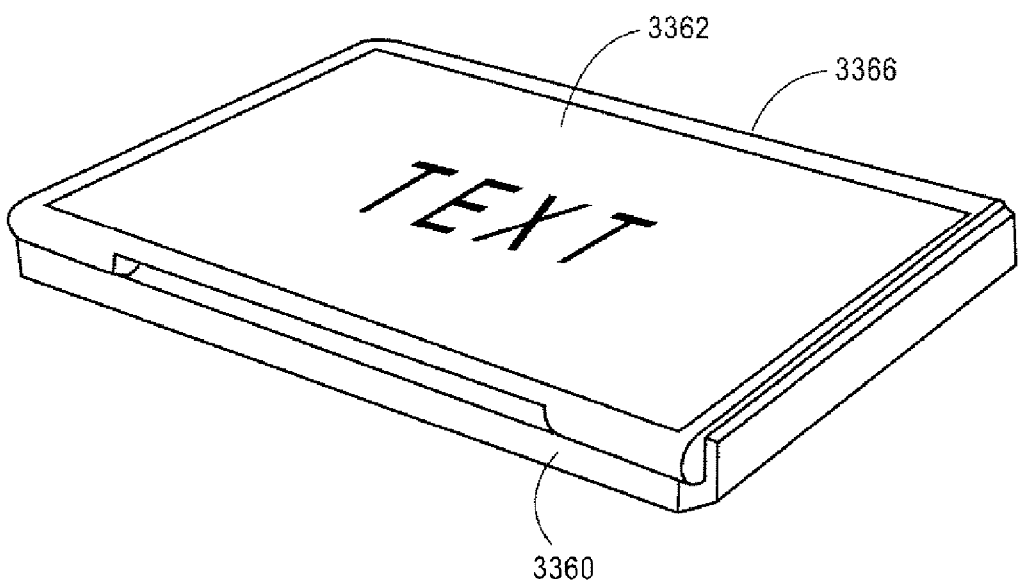


FIG. 33C

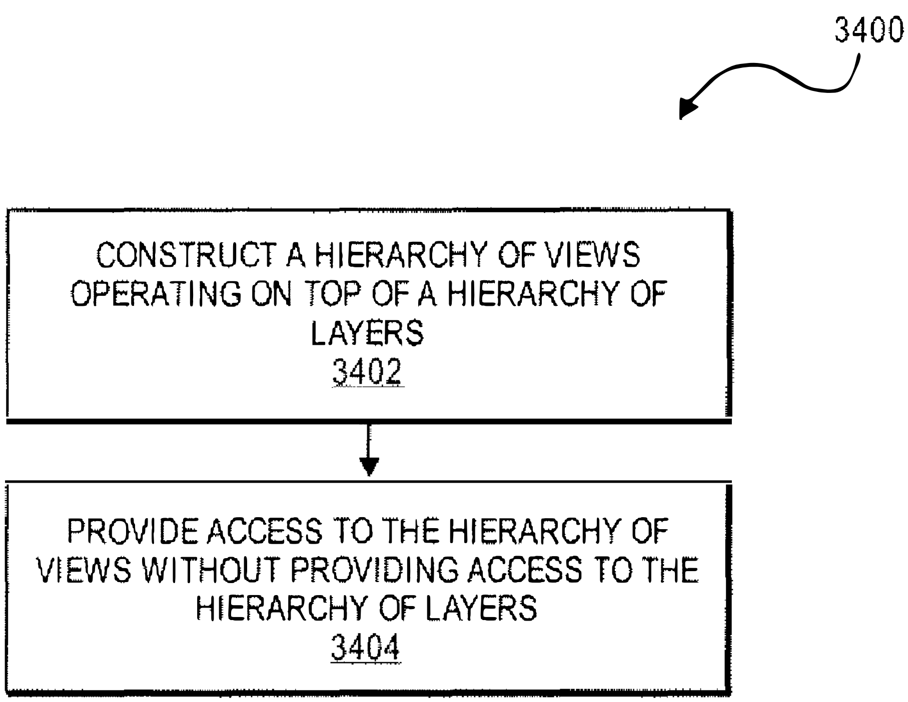

3400
CONSTRUCT A HIERARCHY OF VIEWS OPERATING ON TOP OF A HIERARCHY OF LAYERS
3402
PROVIDE ACCESS TO THE HIERARCHY OF VIEWS WITHOUT PROVIDING ACCESS TO THE HIERARCHY OF LAYERS
3404

FIG. 34

APPLICATION PROGRAMMING INTERFACES FOR SCROLLING OPERATIONS

FIELD OF THE DISCLOSURE

This disclosure relates to application programming interfaces that provide scrolling operations.

COMPUTER PROGRAM LISTING

Applicant has submitted herewith Computer Program Listings which are included as Appendix A, attached.

BACKGROUND OF THE DISCLOSURE

An API is a source code interface that a computer system or program library provides in order to support requests for services from a software application. An API is specified in terms of a programming language that can be interpretative or compiled when an application is built, rather than an explicit low level description of how data is laid out in memory. The software that provides the functionality described by an API is said to be an implementation of the API.

Various devices such as electronic devices, computing systems, portable devices, and handheld devices have software applications. The API interfaces between the software applications and user interface software to provide a user of the device with certain features and operations. A user may desire certain operations such as scrolling, selecting, gesturing, and animating operations for a display of the device.

Scrolling is the act of sliding a directional (e.g., horizontal or vertical) presentation of content, such as text, drawings, or images, across a screen or display window. In a typical graphical user interface, scrolling is done with the help of a scrollbar or using keyboard shortcuts often the arrow keys. Gesturing is a type of user input with two or more input points. Animating operations include changing content within a given time period.

The various types of devices may have a limited display size, user interface, software, API interface and/or processing capability which limits the ease of use of the devices. User interfaces of devices implement APIs in order to provide requested functionality and features. These user interfaces can have difficulty interpreting the various types of user inputs and providing the intended functionality associated with the user inputs.

SUMMARY OF THE DESCRIPTION

At least certain embodiments of the present disclosure include one or more application programming interfaces in an environment with user interface software interacting with a software application. Various function calls or messages are transferred via the application programming interfaces between the user interface software and software applications. Example application programming interfaces transfer function calls to implement scrolling, gesturing, and animating operations for a device.

At least certain embodiments of the present disclosure include an environment with user interface software interacting with a software application. A method for operating through an application programming interface (API) in this environment includes transferring a set bounce call. The method further includes setting at least one of maximum and minimum bounce values. The set bounce call causes a bounce of a scrolled region in an opposite direction of a scroll based on a region past an edge of the scrolled region being visible in a display region at the end of the scroll.

At least certain embodiments of the present disclosure include an environment with user interface software interacting with a software application. A method for operating through an application programming interface (API) in this environment includes transferring a rubberband call. Rubberbanding a scrolled region within a display region occurs by a predetermined maximum displacement when the scrolled region exceeds a display edge. The method further includes transferring an edge rubberband call to set displacement values for at least one edge of the display (e.g., top and bottom edges, left and right edges).

At least certain embodiments of the present disclosure include gesture operations for a display of a device. The gesture operations include performing a scaling transform such as a zoom in or zoom out in response to a user input having two or more input points. The gesture operations also include performing a rotation transform to rotate an image or view in response to a user input having two or more input points.

At least certain embodiments of the present disclosure include a method for performing animations for a display of a device. The method includes starting at least one animation. The method further includes determining the progress of each animation. The method further includes completing each animation based on a single timer. The single timer can be based on a redraw interval of the display hardware.

Various devices which perform one or more of the foregoing methods and machine readable media which, when executed by a processing system, cause the processing system to perform these methods, are also described.

Other methods, devices and machine readable media are also described.

BRIEF DESCRIPTION OF THE DRAWINGS

The disclosure is described by way of example with reference to the accompanying drawings, wherein:

FIG. **1** is flow chart of a method for responding to a user input of a data processing device;

FIG. **2** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure;

FIG. **3** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure;

FIG. **4** is a schematic diagram illustrating an embodiment of user interface of a portable electronic device **400** having a touch-sensitive display **408**;

FIG. **5**A-**5**C illustrate at least some embodiments of user interface of a portable electronic device **400** having a touch-sensitive display;

FIG. **6**A-**6**D illustrate the scrolling of a list of items to a terminus of the list, at which point an area beyond the terminus is displayed and the list is then scrolled in an opposite direction until the area beyond the terminus is no longer displayed, in accordance with some embodiments;

FIG. **7** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure;

FIG. **8** illustrates first and second scroll angles for locking a scroll of a display of a device in a horizontal or vertical direction according to certain teachings of the present disclosure;

FIG. **9** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure;

FIG. **10** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure;

FIG. **11** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure;

FIG. **12** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure;

FIG. **13** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure;

FIG. **14** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure;

FIG. **15** illustrates a display of a device having a scaling transform of a view:

FIGS. **16**A and **16**B illustrate a display of a device with a view having a first and a second scaling factor;

FIG. **16**C illustrates changing a view from a scale factor of 2× to a scale factor of 1× in at least some embodiments of the present disclosure;

FIG. **17** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure;

FIG. **18** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure;

FIG. **19** is flow chart of a method for animating views displayed on a display of a device;

FIG. **20** is flow chart of a method for animating views displayed on a display of a device;

FIG. **21** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure;

FIGS. **22**A and **22**B illustrate synchronizing the resizing of windows of a display of a device;

FIG. **23** illustrates a method for switching ownership of a view of an application displayed on a display of a data processing device;

FIG. **24** illustrates a method for memory management of a view of an application displayed on a display of a device;

FIGS. **25**A and **258** illustrate a data structure having a hierarchy of layers with a layer being associated with a view;

FIG. **26** illustrates a method for compositing media and non-media content of user interface for display on a device;

FIG. **27** illustrates a data structure or layer tree having a hierarchy of layers;

FIG. **28** is a perspective view of a device in accordance with one embodiment of the present disclosure;

FIG. **29** is a perspective view of a device in accordance with one embodiment of the present disclosure;

FIGS. **30**A and **30**B illustrate a device **3070** according to one embodiment of the disclosure;

FIG. **31** is a block diagram of a system in which embodiments of the present disclosure can be implemented;

FIG. **32** shows another example of a device in accordance with one embodiment of the present disclosure;

FIG. **33**A is a perspective view of a device in a first configuration (e.g. in a laptop configuration) in accordance with one embodiment of the present disclosure;

FIG. **33**B is a perspective view of the device of FIG. **33**A in a second configuration (e.g. a transition configuration) in accordance with one embodiment of the present disclosure;

FIG. **33**C is a perspective view of the device of FIG. **33**A in a third configuration (e.g., a tablet configuration) in accordance with one embodiment of the present disclosure; and

FIG. **34** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure.

DETAILED DESCRIPTION

Various embodiments and aspects of the disclosure will be described with reference to details discussed below, and the accompanying drawings will illustrate the various embodiments. The following description and drawings are illustrative of the disclosure and are not to be construed as limiting the disclosure. Numerous specific details are described to provide a through understanding of various embodiments of the present disclosure. However, in certain instances, well-known or conventional details are not described in order to provide a concise discussion of embodiments of the present disclosure.

Some portions of the detailed descriptions which follow are presented in terms of algorithms which include operations on data stored within a computer memory. An algorithm is generally a self-consistent sequence of operations leading to a desired result. The operations typically require or involve physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, can refer to the action and processes of a data processing system, or similar electronic device, that manipulates and transforms data represented as physical (electronic) quantities within the system's registers and memories into other data similarly represented as physical quantities within the system's memories or registers or other such information storage, transmission or display devices.

The present disclosure can relate to an apparatus for performing one or more of the operations described herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a machine (e.g. computer) readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), erasable programmable ROMs (EPROMs), electrically erasable programmable ROMs (EEPROMs), flash memory, mag-

netic or optical cards, or any type of media suitable for storing electronic instructions, and each coupled to a bus.

A machine-readable medium includes any mechanism for storing information in a form readable by a machine (e.g., a computer). For example, a machine-readable medium includes read only memory ("ROM"); random access memory ("RAM"); magnetic disk storage media; optical storage media; flash memory devices; etc.

At least certain embodiments of the present disclosure include one or application programming interfaces in an environment with user interface software interacting with a software application. Various function calls or messages are transferred via the application programming interfaces between the user interface software and software applications. Transferring the function calls or messages may include issuing, initiating, invoking or receiving the function calls or messages. Example application programming interfaces transfer function calls to implement scrolling, gesturing, and animating operations for a device having a display region. An API may also implement functions having parameters, variables, or pointers. An API may receive parameters as disclosed or other combinations of parameters. In addition to the APIs disclosed, other APIs individually or in combination can perform similar functionality as the disclosed APIs.

The display region is a form of a window. A window is a display region which may not have a border and may be the entire display region or area of a display. In some embodiments, a display region may have at least one window and/or at least one view (e.g., web, text, or image content). A window may have at least one view. The methods, systems, and apparatuses disclosed can be implemented with display regions, windows, and/or views.

At least certain embodiments of the present disclosure include scrolling operations for scrolling a display of a device. The scrolling operations include bouncing a scrolled region in an opposite direction of a scroll when a scroll completes, rubberbanding a scrolled region by a predetermined maximum displacement when the scrolled region exceeds a display edge, and setting a scrolling angle that locks the scroll in a horizontal or vertical direction.

At least certain embodiments of the present disclosure include gesture operations for a display of a device. The gesture operations include performing a scaling transform such as a zoom in or zoom out in response to a user input having two or more input points. The gesture operations also include performing a rotation transform to rotate an image or view in response to a user input having two or more input points.

At least certain embodiments of the present disclosure include a method for performing animations for a display of a device. The method includes starting at least one animation. The method further includes determining the progress of each animation. The method further includes completing each animation based on a single timer. The single timer can be based on a redraw interval of the display hardware.

At least certain embodiments of the disclosure may be part of a digital media player, such as a portable music and/or video media player, which may include a media processing system to present the media, a storage device to store the media and may further include a radio frequency (RF) transceiver (e.g., an RF transceiver for a cellular telephone) coupled with an antenna system and the media processing system. In certain embodiments, media stored on a remote storage device may be transmitted to the media player through the RF transceiver. The media may be, for example, one or more of music or other audio, still pictures, or motion pictures.

The portable media player may include a media selection device, such as a click wheel input device on an iPod® or iPod Nano® media player from Apple Computer, Inc. of Cupertino, Calif., a touch screen input device, pushbutton device, movable pointing input device or other input device. The media selection device may be used to select the media stored on the storage device and/or the remote storage device. The portable media player may, in at least certain embodiments, include a display device which is coupled to the media processing system to display titles or other indicators of media being selected through the input device and being presented, either through a speaker or earphone(s), or on the display device, or on both display device and a speaker or earphone(s). In some embodiments, the display device and input device are integrated while in other embodiments the display device and input device are separate devices. Examples of a portable media player are described in published U.S. patent application Nos. 2003/0095096 and 2004/0224638, both of which are incorporated by reference.

Embodiments of the disclosure described herein may be part of other types of data processing systems, such as, for example, entertainment systems or personal digital assistants (PDAs), or general purpose computer systems, or special purpose computer systems, or an embedded device within another device, or cellular telephones which do not include media players, or multi touch tablet devices, or other multi touch devices, or devices which combine aspects or functions of these devices (e.g., a media player, such as an iPod®, combined with a PDA, an entertainment system, and a cellular telephone in one device). In this disclosure, electronic devices and consumer devices are types of devices.

FIG. **1** is flow chart of a method for responding to a user input of a device. The method **100** includes receiving a user input at block **102**. The user input may be in the form of an input key, button, wheel, touch, or other means for interacting with the device. The method **100** further includes creating an event object in response to the user input at block **104**. The method **100** further includes determining whether the event object invokes a scroll or gesture operation at block **106**. For example, a single touch that drags a distance across a display of the device may be interpreted as a scroll operation. In one embodiment, a two or more finger touch of the display may be interpreted as a gesture operation. In certain embodiments, determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period. The method **100** further includes issuing at least one scroll or gesture call based on invoking the scroll or gesture operation at block **108**. The method **100** further includes responding to at least one scroll call, if issued, by scrolling a window having a view (e.g., web, text, or image content) associated with the event object based on an amount of a scroll with the scroll stopped at a predetermined position in relation to the user input at block **110**. For example, an input may end at a certain position on a display of the device. The scrolling may continue until reaching a predetermined position in relation to the last input received from the user. The method **100** further includes responding to at least one gesture call, if issued, by changing a view associated with the event object based on receiving a plurality of input points in the form of the user input at block **112**.

In certain embodiments of the present disclosure scroll operations include attaching scroll indicators to a content edge of a display. Alternatively, the scroll indicators can be attached to the display edge. In some embodiments, user input in the form of a mouse/finger down causes the scroll indicators to be displayed on the display edge, content edge, or window edge of the scrolled region. If a mouse/finger up is

then detected, the scroll indicators are faded out from the display region, content edge, or window edge of the scrolled region.

In certain embodiments of the present disclosure, gesture operations include responding to at least one gesture call, if issued, by rotating a view associated with the event object based on receiving a plurality of input points in the form of the user input. Gesture operations may also include scaling a view associated with the event object by zooming in or zooming out based on receiving the user input.

In some embodiments, a device includes a display region having multiple views or windows. Each window may have a multiple views including superviews and subviews. It is necessary to determine which window, view, superview, or subview is contacted by a user input in the form of a mouse up, mouse down, or drag, etc. An API can set various modes for making this determination. In one embodiment, a pass mode sends mouse down, mouse up, and drag inputs to the nearest subview. In another embodiment, an intercept on drag mode sends a drag input to the superview while mouse up and down inputs are sent to the subview. In another embodiment, an intercept mode sends all drag, mouse up and down inputs to the superview. The superview may be scroller software operating as a subclass of a view software. The subview may be view software operating as a subclass of the user interface software.

FIG. **2** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure. The application programming interface operates in an environment with user interface software interacting with a software application in order to provide a bounce operation. The method **200** for providing a bounce operation includes transferring a set bounce call at block **202**. The method **200** further includes setting at least one of maximum and minimum bounce values at block **204**. The minimum and maximum bounce values may be associated with at least one edge of a window that has received a user input. The method **200** further includes causing a bounce of a scrolled region in an opposite direction of a scroll based on a region past the scrolled region being visible in a display region at the end of the scroll at block **206**. The scrolled region may be a content region.

In certain embodiments of the present disclosure, transferring the set bounce call is either one of issuing, initiating, invoking or receiving the set bounce call.

FIG. **3** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure. The application programming interface operates in an environment with user interface software interacting with a software application in order to provide a rubberband operation. The method **300** for providing a rubberband operation includes transferring a rubberband call to cause rubberbanding a scrolled region displayed within a display at block **302**. The method **300** further includes transferring an edge rubberband call to set displacement values for at least one edge of the display at block **304**. In some embodiments, the displacement values are set for top and bottom edges, left and right edges, or all edges.

Rubberbanding a scrolled region according to the method **300** occurs by a predetermined maximum displacement value when the scrolled region exceeds a display edge of a display of a device based on the scroll. If a user scrolls content of the display making a region past the edge of the content visible in the display, then the displacement value limits the maximum amount for the region outside the content. At the end of the scroll, the content slides back making the region outside of the content no longer visible on the display.

In certain embodiments of the present disclosure, transferring the rubberband call is either one of issuing, initiating, invoking or receiving the rubberband call.

FIG. **4** is a schematic diagram illustrating an embodiment of user interface of a portable electronic device **400** having a touch-sensitive display **408**. The display **408** may include a window **410**. The window **410** may include one or more displayed objects, such as information objects **412-1** to **412-4**. In an exemplary embodiment, the information objects **412** may correspond to contact information for one or more individuals in a list of items. The displayed objects may be moved in response to detecting or determining movement **414** of a point of contact with the display, such as that associated with one or more digits **416** of a user (which are not drawn to Scale in FIG. **4**). In some embodiments, movement of the displayed objects may be accelerated in response to detecting or determining accelerated movement of the point of contact. While embodiment **400** includes one window **410**, in other embodiments there may be two or more display windows. In addition, while embodiment **400** illustrates movement **414** in a particular direction, in other embodiments movement of the displayed objects may be in response to movement **414** in one or more other directions, or in response to a scalar (i.e., a determined or detected movement independent of the direction).

FIGS. **5**A-**5**C illustrate the scrolling of a list of items on a device to a terminus of the list, at which point one or more displayed items at the end of the list smoothly bounce off the end of the display, reverse direction, and then optionally come to a stop. FIG. **5**A is a schematic diagram illustrating an embodiment of user interface of a portable electronic device **400** having a touch-sensitive display. One or more displayed objects, such as information object **412-1** may be a distance **512-1** from a terminus **514** of the list of items which is an edge of a scrolled region and may be moving with a velocity **510-1** while the list is being scrolled. Note that the terminus **514** is a virtual boundary associated with the displayed objects, as opposed to a physical boundary associated with the window **410** and/or the display **408**. As illustrated in FIG. **5**B, when the one or more displayed objects, such as the information object **412-1**, reach or intersect with the terminus **514**, the movement corresponding to the scrolling may stop, i.e., the scrolling velocity may be zero at an instant in time. As illustrated in FIG. **5**C, the one or more displayed objects, such as the information **412-1**, may subsequently reverse direction. At a time after the intersection with the terminus **514**, the information object **412-1** may have velocity **510-2** and may be a distance **512-2** from the terminus **514**. In some embodiments, the magnitude of velocity **510-2** may be less than the magnitude of velocity **510-1** when the distance **512-2** equals the distance **512-1**, i.e., the motion of the one or more displayed objects is damped after the scrolling list reaches and "bounces" at its terminus.

In at least some embodiments of the present disclosure, the method **200** performs the bounce operations described in FIGS. **5**A-**5**C. The bounce call transferred at block **202** determines whether a bounce operation is enabled. The maximum and minimum bounces values determine the amount of bouncing of the scrolled region in an opposite direction of the scroll.

FIGS. **6**A-**6**D illustrate the scrolling of a list of items to a terminus of the list, at which point an area beyond the terminus is displayed and the list is then scrolled in an opposite direction until the area beyond the terminus is no longer displayed, in accordance with some embodiments. The rubberband operation of method **300** is illustrated in the example of FIGS. **6**A-**6**D with the listed items being email messages.

FIGS. **6**A-**6**D illustrate an exemplary user interface **3500**A for managing an inbox in accordance with some embodiments. An analogous user interface may be used to display and manage other mailboxes (e.g., drafts, sent, trash, personal, etc.). In addition, other types of lists are possible, including but not limited to lists of instant message conversations, favorite phone numbers, contact information, labels, email folders, email addresses, physical addresses, ringtones, or album names.

If the list of emails fills more than the allotted screen area, the user may scroll through the emails using vertically upward and/or vertically downward swipe gestures on the touch screen. In the example of FIG. **6**A, a portion of a list of emails is displayed in the screen area, including a top displayed email **3530** from Bruce Walker and a bottom displayed email **3532** from Kim Brook. A user performs a vertically downward swipe gesture **3514** to scroll toward the top of the list. The vertically downward gesture **3514** need not be exactly vertical; a substantially vertical gesture is sufficient. In some embodiments, a gesture within a predetermined angle of being perfectly vertical results in vertical scrolling.

As a result of detecting the vertically downward gesture **3514**, in FIG. **6**B the displayed emails have shifted down, such that the previous bottom displayed email **3532** from Kim Brook is no longer displayed, the previous top displayed email **3530** from Bruce Walker is now second from the top, and the email **3534** from Aaron Jones, which was not displayed in FIG. **6**A, is now displayed at the top of the list.

In this example, the email **3534** from Aaron Jones is the first email in the list and thus is the terminus of the list. Upon reaching this email **3534**, in response to continued detection of the vertically downward gesture **3514**, an area **3536** (FIG. **6**C) above the first email **3534** (i.e., beyond the terminus of the list) is displayed. In some embodiments, the area displayed beyond the terminus of the list is visually indistinct from the background of the list. In FIG. **6**C, both the area **3536** and the background of the emails (e.g., emails **3534** and **3530**) are white and thus are visually indistinct.

Once vertically downward gesture **3514** is complete, such that a corresponding object is no longer detected on or near the touch screen display, the list is scrolled in an opposite direction until the area **3536** is no longer displayed. FIG. **6**D illustrates the result of this scrolling in the opposite direction, the email **3534** from Aaron Jones is now displayed at the top of the screen area allotted to the list and the area **3536** is not displayed.

In the example of FIGS. **6**A-**6**D, a vertically downward gesture resulted in display of an area beyond the first item in the list. As described in FIG. **3**, the values for the predetermined maximum displacement (e.g., display of an area beyond the first item in the list) are set at block **304** for top and bottom edges or at block **306** for all edges of the window.

Similarly, a vertically upward gesture may result in display of an area beyond the last item of the list, if the vertically upward gesture continues once the list has been scrolled to the last item. The last item may be considered a terminus of the list, similar to the first item. As discussed above, the gesture need not be exactly vertical to result in vertical scrolling; a gesture within a predefined range of angles from perfectly vertical is sufficient.

FIG. **7** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure. The application programming interface operates in an environment with user interface software interacting with a software application in order to provide a directional scrolling operation. The method **700** for operating through an application programming interface (API) includes transferring a directional scroll angle call to determine if directional scrolling is enabled at block **702**. The method **700** further includes transferring a directional scroll angle call to set a scroll angle for locking the scrolling in at least one of a vertical or a horizontal direction at block **704**. The method **700** further includes locking the scrolling in the horizontal direction if a user input forms an angle with a horizontal direction that is less than or equal to a first scroll angle at block **706**. The method **700** further includes locking the scrolling in the vertical direction if a user input forms an angle with the vertical direction that is less than or equal to a second scroll angle at block **708**.

In certain embodiments, a user input in the form of a drag forms an angle with the horizontal direction that is less than the first scroll angle. In this case, the user presumably intends to scroll in the horizontal direction. The scrolling will be locked in the horizontal direction until the user input exceeds the first scroll angle. A second scroll angle may be used for locking the user input in the vertical direction. The second scroll angle may be set equal to the first scroll angle.

FIG. **8** illustrates first and second scroll angles for locking a scroll of a display of a device in a horizontal or vertical direction. The horizontal direction **802** and vertical direction **804** are in reference to a window or a display of a device. As discussed in the method **700**, a user input such as a drag movement forming an angle with the horizontal direction **802** less than or equal to the first scrolling angle **806** or **808** will lock the user input in the horizontal direction. In a similar manner, a user input forming an angle with the vertical direction **810** less than or equal to the second scrolling angle **810** or **812** will lock the user input in the vertical direction. The first and second scrolling angles may be set at the same angle or at different angles as well. For example, the first and second scrolling angles may be set at 25 degrees. A user input less than or equal to 25 degrees with respect to the horizontal or vertical direction will lock the scrolling in the appropriate direction.

In some embodiments, the horizontal and vertical locking angles can be determined in part by the aspect of the content. For example, content in the form of a tall page may receive a larger vertical locking angle compared to the horizontal locking angle.

FIG. **9** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure. The application programming interface operates in an environment with user interface software interacting with a software application in order to provide a deceleration scroll operation. The method **900** for providing the deceleration scroll operation includes transferring a deceleration scroll call to set a deceleration factor for a drag user input at block **902**. The method **900** further includes slowing the scroll to a stop based on the speed of the drag user input and the deceleration factor at block **904**.

In certain embodiments, a user input in the form of a drag invokes a scroll operation for a certain time period. The user input has a certain speed. The scroll of the scrolled region of a window or a display region of a display of a device will be stopped after the user input stops by applying a deceleration factor to the speed of the user input during the drag movement.

FIG. **10** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure. The application programming interface operates in an environment with user interface software interacting with a software application in order to provide a scroll hysteresis operation. The method **1000** for providing the scroll hysteresis operation includes transferring a scroll

hysteresis call to determine whether a user input invokes a scroll at block **1002**. The method **1000** further includes setting a hysteresis value for determining whether a user input invokes a scroll at block **1004**.

In certain embodiments, a user input in the form of a drag over a certain distance across a display or window within a display of a device invokes a scroll operation. The hysteresis value determines the certain distance which the user input must drag across the display or window prior to invoking a scroll operation. A user input that does not drag the certain predetermined distance will not invoke a scroll operation and may be considered a mouse up or down input or other type of input.

FIG. **11** illustrates details of all application programming interface in flow chart form according to certain teachings of the present disclosure. The application programming interface operates in an environment with user interface software interacting with a software application in order to attach a scroll indicator to a scroll region edge or a window edge of a device. In some embodiments, the scroll region edge is associated with a content edge. The window or display edge may be associated with the edge of a display region. The method **1100** for providing the scroll indicator includes transferring a scroll indicator call to determine whether at least one scroll indicator attaches to an edge of a scroll region or a window edge at block **1102**. A scroll indicator may be displayed on any display edge, window edge or scroll region edge. The method **1100** further includes optionally attaching at least one scroll indicator to the edge of the scroll region based on the scroll indicator call at block **1104**. Alternatively, the method **1100** further includes optionally attaching at least one scroll indicator to the window edge of the view based on the scroll indicator call at block **1106**.

In some embodiments, the operations of method **1100** can be altered, modified, combined, or deleted. For example, block **1104** can be deleted. Likewise, block **1106** can be deleted from the method **1100**. Alternatively, the order of block **1104** and block **1106** can be switched. Other methods having various operations that have been disclosed within the present disclosure can also be altered, modified, rearranged, collapsed, combined, or deleted.

In certain embodiments of the present disclosure, transferring the scroll indicator call is either one of issuing, initiating, invoking or receiving the scroll indicator call. For example, the user interface software (e.g., software kit or library) may receive the scroll indicator call from the software application.

FIG. **12** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure. The application programming interface operates in an environment with user interface software interacting with a software application in order to determine if an inadvertent user input contacts a view of a display of a device. The method **1200** includes transferring an inadvertent user input call to determine whether the user input was inadvertent at block **1202**. The method **1200** further includes ignoring the inadvertent user input based on the determination of the inadvertent user input call at block **1204**. In one embodiment, the inadvertent user input call comprises a thumb detection call to determine whether the user input was an inadvertent thumb.

In certain embodiments of the present disclosure, transferring the inadvertent user input call is either one of issuing, initiating, invoking or receiving the inadvertent user input call.

A gesture API provides an interface between an application and user software in order to handle gesturing. Gesturing may include scaling, rotating, or other changes to a view, window, or display. A mask may merely permit certain changes while limiting or not permitting other changes. Events of all kinds come into the application via a graphics framework. They are enqueued, collaleced if necessary and dispatched. If the events are system level events (e.g., application should suspend, device orientation has chanted, etc) they are routed to the application having an instance of a class of the user interface software. If the events are hand events based on a user input, the events are routed to the window they occurred over. The window then routes these events to the appropriate control by calling the instance's mouse and gesture methods. The control that receives a mouse down or mouse entered function will continue to get all future calls until the hand is lifted. If a second finger is detected, the gesture methods or functions are invoked. These functions may include start, change, and end gesture calls. The control that receives start gesture call will be sent all future change gesture calls until the gesture ends.

FIG. **13** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure. The application programming interface operates in an environment with user interface software interacting with a software application in order to provide a gesture operation. The method **1300** for providing the gesture operation includes transferring a handle gesture event call at block **1302**. The method **1300** further includes transferring a gesture change call in response to the handle gesture event call at block **1304**.

In certain embodiments, a user input in the form of two or more points is received by a display of a device. A multi-touch driver of the device receives the user input and packages the event into an event object. A window server receives the event object and determines whether the event object is a gesture event object. If the window server determines that a gesture event object has been received, then user interface software issues or transfers the handle gesture call at block **1302** to a software application associated with the view. The software application confirms that a gesture event has been received and passes the handle gesture call to a library of the user interface software. The window server also associates the gesture event object with the view that received the user input. The library responds by transferring a gesture change call in response to the handle gesture event call at block **1304**.

In one embodiment, a window or view associated with the user input receives the change call in order to perform the gesture event. The user software that provides the view receives a gesture start event call, a gesture changed event call, a zoom to scale setting for the view, and a gesture end call. The gesture calls receive an input of a gesture event which may be base event having a type such as a hand event, keyboard event, system event, etc. A delegate associated with the application receives a start gesture call, gesture did change call, and gesture did finish call. The user software is dynamically linking into the application during the run time of the gesture process.

In some embodiments, the gesture changed function call contains the following information about the gesture:

the number of fingers currently down;

the number of fingers initially down;

the rotation of the hand;

the scale of the hand;

the translation of the hand;

the position of the inner and outermost fingers; and

the pressure of the first finger.

In other embodiments, more information about each finger down may be included as follows.

the stage of the finger Oust touch down, fully pressed, lifting off, etc);

the position of the finger;

the proximity of the finger (how hard you're touching);

the orientation of the finger (what angle the ovoid is at);

the length of the major and minor axis,

the velocity of the finger; and

the eccentricity of the finger's ovoid.

A gesture event object may be a chord event object having a chord count (e.g., number of fingers contacted the view or display), a chord start event, a chord change event, and a chord end event. A chord change event may include a scaling or rotation transform.

FIG. **14** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure. The application programming interface operates in an environment with user interface software interacting with a software application in order to provide a scaling transform of a display region, window, or view of a display of a device. The method **1400** for providing the scaling transform includes transferring a scaling transform call to determine a scaling transform for a view associated with a user input having a plurality of input points at block **1402**. The method **1400** further includes transferring a scaling gesture start call at block **1404**. The method **1400** further includes transferring a scaling gesture progress call at block **1406**. The method **1200** further includes transferring a scaling gesture end call at block **1408**.

In certain embodiments, a user input in the form of two or more input points (e.g., fingers) moves together or apart to invoke a gesture event that performs a scaling transform on the view associated with the user input. A scale transform includes a minimum and maximum scale factor. FIG. **15** illustrates a display **1502** of a device having a scaling transform of a view. The view **1504** (e.g., web, text, or image content) has a first scale factor. A user input (e.g., two fingers moving apart) associated with the view **1504** is interpreted as a gesture event to zoom out from view **1504** to view **1508** having a second scale factor that exceeds the maximum scale factor of the view **1516**. A snapback flag determines whether the zoom out can proceed past the maximum scale factor to view **1508** prior to snapping back to the maximum scale factor associated with view **1516**.

FIG. **16**A illustrates a display **1604** of a device having a first scaling factor of a view **1616**. A user input (e.g., two fingers **1608** and **1610** moving together) associated with the view **1614** is interpreted as a gesture event to zoom in from view **1614** to view **1664** having a second scale factor as illustrated in FIG. **16**B. The dashed regions **1602** and **1650** represent the total area of the content with the only content being displayed in the display area **1604** and **1652**. In performing the scaling transform from FIG. **16**A to FIG. **16**B, the center of the gesture event, center **1612** for FIG. **16**A and center **1660** for FIG. **16**B, remains in the same position with respect to the display **1604**. The scroll indicator **1606** shrinks to become scroll indicator **1654** during the transform to indicate that a smaller portion of the total content **1650** is being displayed on display **1604** as a result of the zoom in operation. The dashed region **1650** is larger than the dashed region **1602** to represent that a larger portion of content is not being displayed on display **1652** in FIG. **16**B as a result of the zoom in operation.

In at least some embodiments of the present disclosure, a user desires to change a view **1670** from a scale factor of 2× to a scale factor of 1× as illustrated in FIG. **16**C. A first set of user inputs **1672** and **1674** that move to the second set of user inputs **1676** and **1678** will decrease the scale factor from 2× to 1×. It may be desirable for the user to scale from 2× to 1× without having to move the user inputs a large distance across the view **1670**. In an environment with user interface software interacting with a software application, a gesture scaling transform flag may be set in order to determine a scaling transform for a view associated with a user input having a plurality of input points. The scaling transform flag scales either from a current scale factor to a minimum scale factor or from the current scale factor to a maximum scale factor. For example, a flag may be set at the position associated with a 1.5× scale factor and a third set of user inputs **1680** and **1682**. A user desiring to change the scale factor from 2× to 1× would only have to move his fingers, the user inputs, from the first set **1672** and **1674** to the third set **1680** and **1682** if the gesture scaling transform flag has been set at a scale factor of 1.5×.

FIG. **17** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure. The application programming interface operates in an environment with user interface software interacting with a software application in order to provide a rotation transform of a view, window, or display region of a display of a device. The method **1700** for providing the rotation transform includes transferring a rotation transform call to determine a rotation transform for a view associated with a user input having a plurality of input points at block **1702**. The method **1700** further includes transferring a rotation gesture start call at block **1704**. The method **1700** further includes transferring a scaling gesture progress call at block **1706**. The method **1700** further includes transferring a scaling gesture end call at block **1708**.

In certain embodiments, a user input in the form of two or more input points rotates to invoke a gesture event that performs a rotation transform on the view associated with the user input. The rotation transform includes a minimum and maximum degree of rotation for associated minimum and maximum rotation views. The user input may temporarily rotate a view past a maximum degree of rotation prior to the view snapping back to the maximum degree of rotation.

FIG. **18** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure. The application programming interface operates in an environment with user interface software interacting with a software application in order to notify a delegate of at least one animation associated with a display region, window, or view of a display of a device. A delay in the animation may be specified by the API. Also, multiple animations may be assigned priority by the API. The method **1800** for notifying the delegate includes determining whether any animation occurs at block **1802**. The method **1800** further includes checking the progress of an animation at block **1804**. If progress has occurred, then the next state (e.g., position, opacity, or transform) of the animation can be calculated at block **1806**. If progress has completed at block **1806**, then at block **1808** it is determined whether the view associated with the completed animation is associated with a delegate. If so, a delegate call is transferred to notify the delegate of the animation for the view at block **1810**. The delegate operating

under the control of the software application can change other views in response to the view being modified by the animation.

In certain embodiments, software invokes an animation that performs a scaling transform on the view associated with the user input. A display may include numerous views. The view being increased in size by the scaling transform may obstruct other views in which case the other views may need to be reduced in size. Alternatively, the view being decreased in size by the scaling transform may create additional area for other views to increase in size.

FIG. **19** is flow chart of a method for animating a display region, windows, or views displayed on a display of a device. The method **1900** includes starting at least two animations at block **1902**. The method **1900** further includes determining the progress of each animation at block **1904**. The method **1900** further includes completing each animation based on a single timer at block **1906**.

In certain embodiments of the present disclosure, the single timer includes a timer based on a redraw interval which is a time period between the display of a current frame and a next frame of the display of the device. In this case, changes in animation are updated to the display during the redraw interval in order to display the chances during the next frame of the display. The progress of each animation may be calculated periodically or based upon a progress call.

The method **1900** may further include determining whether each animation is associated with a delegate. The delegate is then notified of the animation. Other views not associated with an animation may be changed depending on the software application controlling the delegate.

FIG. **20** is flow chart of a method for animating a display region, windows, or views displayed on a display of a device. The method **2000** includes providing a single animation timer at block **2002**. The method **2000** further includes animating a plurality of animations with the single animation timer at block **2004**. For example, a single timer may control all animations which occur simultaneously. The animations may include a transform, a frame, and an opacity animation. A animation transform may include a scaling or rotation transform. A frame animation may include resizing of a frame. An opacity animation changes the opacity from opaque to transparent or vice versa.

FIG. **21** illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure. The application programming interface operates in an environment with user interface software interacting with multiple software applications or processes in order to synchronize animations associated with multiple views or windows of a display of a device. The method **2100** for synchronizing the animations includes setting attributes of views independently with each view being associated with a process at block **2102**. For example, an attribute or property of a view may include a position, size, opacity, etc. An animation alters one or more attributes from a first state to a second state. The method **2100** further includes transferring a synchronization call to synchronize animations for the multiple views of the display at block **2104**. The synchronization call may include input parameters or arguments such as an identification of the synchronization of the processes and a list of the processes that are requesting animation of the multiple views. In one embodiment, the synchronization call includes the identification and the number of processes that are requesting animation. In one embodiment, each application or process sends a synchronization call at different times. The method **2100** further includes transferring a synchronization confirmation message when a synchronization flag is enabled at block **2106**. The synchronization flag can be enabled when the processes to be synchronized have each sent messages to a window server operating the user interface software. The method **2100** further includes updating the attributes of the views from a first state to a second state independently at block **2108**. In one embodiment, the window server receives the updated attributes from each process at different times. The method **2100** further includes transferring a start animation call to draw the requested animations when both processes have updated attributes associated with the second state at block **2110**.

In some embodiments, a first data structure or layer tree represents a hierarchy of layers that correspond to the views or windows of the processes. A second data structure or render tree represents a similar copy of the layer tree. However, the render tree is not updated until the independent processes have completed their separate animations. At this time, the render tree updates and redraws the screen with the new animations.

FIGS. **22**A and **22**B illustrate synchronizing the resizing of views or windows of a display of a device. For example, a window **2210** associated with a first process with a size attribute may increase in size by changing from a first state, window **2210** in FIG. **22**A, to a second state window **2210** in FIG. **22**B. At approximately the same time, a second window **2220** may decrease in size in proportion to the increase in size of the first window **2210**. The method **2100** provides synchronization of the resizing of the windows **2210** and **2220** illustrated in FIGS. **22**A and **22**B. The animations in changing from the first state to the second state may occur incrementally and occur with the synchronization of method **2100**.

FIG. **23** illustrates a method for switching ownership of a view of an application displayed on a display of a data processing device. The method **2300** includes constructing a data structure having a hierarchy of layers with a layer being associated with a view and owning the view at block **2302**. The layers may be content, windows, video, images, text, media, or any other type of object for user interface of the application. The method **2300** further includes removing the layer from the data structure at block **2304**. The method **2300** further includes switching ownership of the view from the layer to the view at block **2306**.

In some embodiments, each layer from the data structure is associated with a view. The layer associated with the view sends a delegate function call to the view in order to generate content provided by the view. A first pointer reference points from the layer to the view. A second pointer reference points from the view to the layer. The number of references pointing to an object such as the view is defined as the retained count of the object. The view may receive notification that the layer will be removed from the data structure. Removing the layer from the data structure may occur based on the view associated with the layer being removed from the display of the device. When the layer is removed from the data structure or layer tree the pointer from the layer to the view will be removed. The view will have a retained count of zero and be deallocated or removed from memory if the ownership of the view is not reversed. The view will have a retained count of at least one if ownership is reversed.

FIG. **24** illustrates a method for memory management of a view of an application displayed on a display of a device. The method **2400** includes constructing a data structure having a hierarchy of layers with at least one layer being associated with the view at block **2402**. The method **2400** further includes storing the data structure in memory at block **2404**. The method **2400** further includes maintaining a retained count of the number of references to the view from other

objects at block **2406**. The method **2400** further includes deallocating the view from memory if the retained count is zero at block **2408**. As discussed above, the retained count of the view will be decremented if the layer is removed from the data structure. Removing the layer from the data structure may occur based on the view associated with the layer being removed from the display of the device.

FIGS. **25**A and **25**B illustrate a data structure having a hierarchy of layers with a layer being associated with a view. The data structure includes layers **2502**, **2504**, and **2506**. Layer **2506** is associated with the view **2510**. The layer **2506** associated with the **2510** view sends a delegate call to the view in order to generate content provided by the view. A first pointer reference **2508** points from the layer **2506** to the view **2510**. A second pointer reference **2512** points from the view **2510** to the layer **2506**. A third pointer reference **2532** may point from user interface (UI) controller **2530** to the view **2510**. The UI controller **2530** may control operations associated with the view **2510** such as scrolling the view **2510** in response to a user input. The view **2510** in FIG. **25**A has a retained count of two based on the pointer references **2508** and **2532**.

If the layer **2506** is removed from the data structure as illustrated in FIG. **25**B, then the pointer **2508** is removed. View **2510** will have a lower retained count as illustrated in FIG. **25**B. If view **2510** has a retained count of zero, then the memory storing the view **2510** will be deallocated.

FIG. **26** illustrates a method for compositing media and non-media content of user interface for display on a device. The method **2600** includes constructing a data structure having a hierarchy of layers associated with the user interface of the device at block **2602**. The method **2600** further includes determining whether each layer of the data structure is associated with media or non-media content at block **2604**. The data structure or layer tree is traversed in order to determine whether each of the layers of the data structure is associated with media or non-media content. The method **2600** further includes detaching a layer associated with media content from the data structure at block **2606**. The method **2600** further includes storing media content in a first memory location at block **2606**. The method **2600** further includes storing non-media content in a second memory location at block **2608**. The method **2600** further includes compositing the media and non-media content for display on the device at block **2610**.

In some embodiments, compositing the media and non-media content includes retrieving the media content from the first memory location, retrieving the non-media content from the second memory location, and scanning the media and non-media content directly to the display. The memory location can be any type of memory located in cache, main memory, a graphics processing unit, or other location within a device. The media content may include video, video plug-in, audio, image, or other time varying media. The media content may be in the form of a YUV model with the Y representing a luminance component (the brightness) and U and V representing chrominance (color) components. The media content may be scanned to the display at a rate of substantially twenty to forty frames per second. The media content may be scaled prior to being scanned to the display of the device.

The non-media content may include content, views, and images that do not require frequent updating. The non-media content may be in the form of a RGB model which is an additive model in which red, green, and blue (often used in additive light models) are combined in various ways to reproduce other colors. The non-media content may be scanned to the display at a slower rate compared to the media content.

FIG. **27** illustrates a data structure or layer tree having a hierarchy of layers. The layers can be associated with media and non-media content. For example, layer **2704** is associated with media content **2706** such as a video. Layer **2710** is associated with non-media content **2712** which may be user interface view for the video. Layers **2720**, **2730**, and **2740** are associated with non-media content **2722**, **2732**, and **2742**, respectively, that forms the components of the non-media content **2712**. The method **2600** will determine whether each layer of the data structure is associated with media or non-content. Any layers associated with media content such as layer **2704** will be removed from the data structure and processed in a separate memory location.

In some embodiments, the methods, systems, and apparatuses of the present disclosure can be implemented in various devices including electronic devices, consumer devices, data processing devices, desktop computers, portable computers, wireless devices, cellular devices, tablet devices, handheld devices, multi touch devices, multi touch data processing devices, any combination of these devices, or other like devices. FIGS. **4-6** and **28-33** illustrate examples of a few of these devices.

FIG. **28** illustrates a device **2800** according to one embodiment of the disclosure. FIG. **28** shows a wireless device in a telephone configuration having a "candy-bar" style. In FIG. **28**, the wireless device **2800** may include a housing **2832**, a display device **2834**, an input device **2836** which may be an alphanumeric keypad, a speaker **2838**, a microphone **2840** and an antenna **2842**. The wireless device **2800** also may include a proximity sensor **2844** and an accelerometer **2846**. It will be appreciated that the embodiment of FIG. **28** may use more or fewer sensors and may have a different form factor from the form factor shown in FIG. **28**.

The display device **2834** is shown positioned at an upper portion of the housing **2832**, and the input device **2836** is shown positioned at a lower portion of the housing **2832**. The antenna **2842** is shown extending from the housing **2832** at an upper portion of the housing **2832**. The speaker **2838** is also shown at an upper portion of the housing **2832** above the display device **2834**. The microphone **2840** is shown at a lower portion of the housing **2832**, below the input device **3286**. It will be appreciated that the speaker **2838** and microphone **2840** can be positioned at any location on the housing, but are typically positioned in accordance with a user's ear and mouth, respectively.

The display device **2834** may be, for example, a liquid crystal display (LCD) which does not include the ability to accept inputs or a touch input screen which also includes an LCD. The input device **2836** may include, for example, buttons, switches, dials, sliders, keys or keypad, navigation pad, touch pad, touch screen, and the like. Any well-known speaker, microphone and antenna can be used for speaker **2838**, microphone **2840** and antenna **2842**, respectively.

The data acquired from the proximity sensor **2844** and the accelerometer **2846** can be combined together, or used alone, to gather information about the user's activities. The data from the proximity sensor **2844**, the accelerometer **2846** or both can be used, for example, to activate/deactivate a display backlight, initiate commands, make selections, control scrolling, gesturing, animating or other movement in a display, control input device settings, or to make other changes to one or more settings of the device. In certain embodiments of the present disclosure, the device **2800** can be used to implement at least some of the methods discussed in the present disclosure.

FIG. **29** shows a device **2950** in accordance with one embodiment of the disclosure. The device **2950** may include a housing **2952**, a display/input device **2954**, a speaker **2956**, a microphone **2958** and an optional antenna **2960** (which may be visible on the exterior of the housing or may be concealed within the housing). The device **2950** also may include a proximity sensor **2962** and an accelerometer **2964**. The device **2950** may be a cellular telephone or a device which is an integrated PDA and a cellular telephone or a device which is an integrated media player and a cellular telephone or a device which is both an entertainment system (e.g. for playing games) and a cellular telephone, or the device **2950** may be other types of devices described herein. In one particular embodiment, the device **2950** may include a cellular telephone and a media player and a PDA, all contained within the housing **2952**. The device **2950** may have a form factor which is small enough that it fits within the hand of a normal adult and is light enough that it can be carried in one hand by an adult. It will be appreciated that the term "portable" means the device can be easily held in an adult user's hands (one or both); for example, a laptop computer and an iPod are portable devices.

In one embodiment, the display/input device **2954** may include a multi-point touch input screen in addition to being a display, such as an LCD. In one embodiment, the multi-point touch screen is a capacitive sensing medium configured to detect multiple touches (e.g., blobs on the display from a user's face or multiple fingers concurrently touching or nearly touching the display) or near touches (e.g., blobs on the display) that occur at the same time and at distinct locations in the plane of the touch panel and to produce distinct signals representative of the location of the touches on the plane of the touch panel for each of the multiple touches.

In certain embodiments of the present disclosure, the device **2800** can be used to implement at least some of the methods discussed in the present disclosure.

FIGS. **30**A and **30**B illustrate a device **3070** according to one embodiment of the disclosure. The device **3070** may be a cellular telephone which includes a hinge **3087** that couples a display housing **3089** to a keypad housing **3091**. The hinge **3087** allows a user to open and close the cellular telephone so that it can be placed in at least one of two different configurations shown in FIGS. **30**A and **30**B. In one particular embodiment, the hinge **3087** may rotatably couple the display housing to the keypad housing. In particular, a user can open the cellular telephone to place it in the open configuration shown in FIG. **30**A and can close the cellular telephone to place it in the closed configuration shown in FIG. **30**B. The keypad housing **3091** may include a keypad **3095** which receives inputs (e.g. telephone number inputs or other alphanumeric inputs) from a user and a microphone **3097** which receives voice input from the user. The display housing **3089** may include, on its interior surface, a display **3093** (e.g. an LCD) and a speaker **3098** and a proximity sensor **3084**; on its exterior surface, the display housing **3089** may include a speaker **3096**, a temperature sensor **3094**, a display **3088** (e.g. another LCD), an ambient light sensor **3092**, and a proximity sensor **3084**A. Hence, in this embodiment, the display housing **3089** may include a first proximity sensor on its interior surface and a second proximity sensor on its exterior surface.

In at least certain embodiments, the device **3070** may contain components which provide one or more of the functions of a wireless communication device such as a cellular telephone, a media player, an entertainment system, a PDA, or other types of devices described herein. In one implementation of an embodiment, the device **3070** may be a cellular telephone integrated with a media player which plays MP3 files, such as MP3 music files.

Each of the devices shown in FIGS. **4**, **5**A, **5**B, **5**C, **6**A, **6**B, **6**C, **6**D, **28**, **29**, **30**A and **30**B may be a wireless communication device, such as a cellular telephone, and may include a plurality of components which provide a capability for wireless communication. FIG. **31** shows an embodiment of a wireless device **3070** which includes the capability for wireless communication. The wireless device **3070** may be included in any one of the devices shown in FIGS. **4**, **5**A, **5**B, **5**C, **6**A, **6**B, **6**C, **6**D, **28**, **29**, **30**A and **30**B, although alternative embodiments of those devices of FIGS. **4**, **5**A, **5**B, **5**C, **6**A, **6**B, **6**C, **6**D, **28**, **29**, **30**A and **308** may include more or fewer components than the Wireless device **3070**.

Wireless device **3070** may include an antenna system **3101**. Wireless device **3070** may also include a digital and/or analog radio frequency (RF) transceiver **3102**, coupled to the antenna system **3101**, to transmit and/or receive voice, digital data and/or media signals through antenna system **3101**.

Wireless device **3070** may also include a digital processing system **3103** to control the digital RF transceiver and to manage the voice, digital data and/or media signals. Digital processing system **3103** may be a general purpose processing device, such as a microprocessor or controller for example. Digital processing system **3103** may also be a special purpose processing device, such as an ASIC (application specific integrated circuit), FPGA (field-programmable gate array) or DSP (digital signal processor). Digital processing system **3103** may also include other devices, as are known in the art, to interface with other components of wireless device **3070**. For example, digital processing system **3103** may include analog-to-digital and digital-to-analog converters to interface with other components of wireless device **3070**. Digital processing system **3103** may include a media processing system **3109**, which may also include a general purpose or special purpose processing device to manage media, such as files of audio data.

Wireless device **3070** may also include a storage device **3104**, coupled to the digital processing system, to store data and/or operating programs for the Wireless device **3070**. Storage device **3104** may be, for example, any type of solid-state or magnetic memory device.

Wireless device **3070** may also include one or more input devices **3105**, coupled to the digital processing system **3103**, to accept user inputs (e.g., telephone numbers, names, addresses, media selections, etc.) Input device **3105** may be, for example, one or more of a keypad, a touchpad, a touch screen, a pointing device in combination with a display device or similar input device.

Wireless device **3070** may also include at least one display device **33106**, coupled to the digital processing system **3103**, to display information such as messages, telephone call information, contact information, pictures, movies and/or titles or other indicators of media being selected via the input device **3105**. Display device **3106** may be, for example, an LCD display device. In one embodiment, display device **3106** and input device **3105** may be integrated together in the same device (e.g., a touch screen LCD such as a multi-touch input panel which is integrated with a display device, such as an LCD display device). The display device **3106** may include a backlight **3106**A to illuminate the display device **3106** under certain circumstances. It will be appreciated that the Wireless device **3070** may include multiple displays.

Wireless device **3070** may also include a battery **3107** to supply operating power to components of the system including digital RF transceiver **3102**, digital processing system **3103**, storage device **3104**, input device **3105**, microphone

310A, audio transducer 3108, media processing system 3109, sensor(s) 3110, and display device 3106. Battery 3107 may be, for example, a rechargeable or non-rechargeable lithium or nickel metal hydride battery. Wireless device 3070 may also include audio transducers 3108, which may include one or more speakers, and at least one microphone 3105A. In certain embodiments of the present disclosure, the wireless device 3070 can be used to implement at least some of the methods discussed in the present disclosure.

FIG. 32 shows another example of a device according to an embodiment of the disclosure. This device 3200 may include a processor, such as microprocessor 3202, and a memory 3204, which are coupled to each other through a bus 3206. The device 3200 may optionally include a cache 3208 which is coupled to the microprocessor 3202. This device may also optionally include a display controller and display device 3210 which is coupled to the other components through the bus 3206. One or more input/output controllers 3212 are also coupled to the bus 3206 to provide an interface for input/output devices 3214 and to provide an interface for one or more sensors 3216 which are for sensing user activity. The bus 3206 may include one or more buses connected to each other through various bridges, controllers, and/or adapters as is well known in the art. The input/output devices 3214 may include a keypad or keyboard or a cursor control device such as a touch input panel. Furthermore, the input/output devices 3214 may include a network interface which is either for a wired network or a wireless network (e.g. an RF transceiver). The sensors 3216 may be any one of the sensors described herein including, for example, a proximity sensor or an ambient light sensor. In at least certain implementations of the device 3200, the microprocessor 3202 may receive data from one or more sensors 3216 and may perform the analysis of that data in the manner described herein. For example, the data may be analyzed through an artificial intelligence process or in the other ways described herein. As a result of that analysis, the microprocessor 3202 may then automatically cause an adjustment in one or more settings of the device.

In certain embodiments of the present disclosure, the device 3200 can be used to implement at least some of the methods discussed in the present disclosure.

FIGS. 33A-C show another example of a device according to at least certain embodiments of the disclosure. FIG. 33A illustrates a laptop device 3300 with a keyboard 3302, a body 3304, a display frame 3306, and a display 3308. The laptop device 3300 can be converted into a tablet device as illustrated in FIG. 33B and FIG. 33C. FIG. 33B illustrates the conversion of the laptop device into a tablet device. An edge of a display frame 3356 containing a display 3358 is slide within the body 3354 across the top of a keyboard 3352 until forming a tablet device as illustrated in FIG. 33C. The tablet device with a display 2362 and a display frame 3366 rests on top of a body 3360.

In certain embodiments of the present disclosure, the laptop device 3300 can be used to implement at least some of the methods discussed in the present disclosure.

FIG. 34 illustrates details of an application programming interface in flow chart form according to certain teachings of the present disclosure. The application programming interface operates in an environment with user interface software interacting with a software application. In some embodiments, a hierarchy of views operates on top of a hierarchy of layers within the user interface software. The API operates as illustrated in method 3400 that includes constructing a hierarchy of views operating on top of a hierarchy of layers at block 3402. The method 3400 further includes providing access to the hierarchy of views without providing access to

the hierarchy of layers at block 3404. An application may interact with the hierarchy of views via the API without accessing the hierarchy of layers operating below the hierarchy of views.

In some embodiments, a platform provides various scrolling, gesturing, and animating operations. The platform includes hardware components and an operating system. The hardware components may include a processing unit coupled to an input panel and a memory coupled to the processor. The operating system includes one or more programs that are stored in the memory and configured to be executed by the processing unit. One or more programs include various instructions for transferring function calls or messages through an application programming interface in order to perform various scrolling, gesturing, and animating operations.

In an embodiment, the one or more programs include instructions for transferring a bounce call through an API to cause a bounce of a scrolled region in an opposite direction of a scroll based on a region past an edge of the scrolled region being visible in a display region at the end of the scroll. In an embodiment, the one or more programs include instructions for transferring a rubberband call through an API to cause a rubberband effect on a scrolled region by a predetermined maximum displacement when the scrolled region exceeds a display edge based on a scroll. In an embodiment, the one or more programs include instructions for transferring a directional scroll call through an API to set a scroll angle for locking the scrolling in at least one of a vertical or a horizontal direction.

In an embodiment, the one or more programs include instructions for transferring a scroll hysteresis call through an API to determine whether a user input invokes a scroll. In an embodiment, the one or more programs include instructions for transferring a deceleration scroll call through an API to set a deceleration factor for a user input based on the user input invoking a scroll. In an embodiment, the one or more programs include instructions for transferring a scroll indicator call through an API to determine whether at least one scroll indicator attaches to a content edge or a display edge of a display region.

In some embodiments, the platform includes a framework containing a library of software code. The framework interacts with the programs of the platform to provide application programming interfaces for performing various scrolling, gesturing, and animating operations. The framework also includes associated resources (e.g., images, text, etc.) that are stored in a single directory.

In an embodiment, the library of the framework provides an API for specifying a bounce operation to cause a bounce of a scrolled region in an opposite direction of a scroll based on a region past an edge of the scrolled region being visible in a display region at the end of the scroll. In an embodiment, the library of the framework provides an API for specifying a rubberband operation that has a rubberband effect on a scrolled region by a predetermined maximum displacement when the scrolled region exceeds a display edge based on a scroll. In an embodiment, the library of the framework provides an API for specifying a directional scroll operation to set a scroll angle for locking the scrolling in at least one of a vertical or a horizontal direction.

In an embodiment, the library of the framework provides an API for specifying a scroll hysteresis operation to determine whether a user input invokes a scroll. In an embodiment, the library of the framework provides an API for specifying a deceleration scroll operation to set a deceleration factor for a user input based on the user input invoking a scroll. In an

embodiment, the library of the framework provides an API for specifying a scroll indicator operation to determine whether at least one scroll indicator attaches to a content edge or a display edge of a display region.

In the foregoing specification, the disclosure has been described with reference to specific exemplary embodiments thereof. It will be evident that various modifications may be made thereto without departing from the broader spirit and scope of the disclosure as set forth in the following claims. The specification and drawings are, accordingly, to be regarded in an illustrative sense rather than a restrictive sense.

What is claimed is:

1. A machine implemented method for scrolling on a touch-sensitive display of a device comprising:

receiving a user input, the user input is one or more input points applied to the touch-sensitive display that is integrated with the device;

creating an event object in response to the user input;

determining whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation;

issuing at least one scroll or gesture call based on invoking the scroll or gesture operation;

responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object based on an amount of a scroll with the scroll stopped at a predetermined position in relation to the user input; and

responding to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.

2. The method as in claim 1, further comprising:

rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll.

3. The method as in claim 1, further comprising:

attaching scroll indicators to a content edge of the window.

4. The method as in claim 1, further comprising:

attaching scroll indicators to the window edge.

5. The method as in claim 1, wherein determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period.

6. The method as in claim 1, further comprising:

responding to at least one gesture call, if issued, by rotating a view associated with the event object based on receiving a plurality of input points in the form of the user input.

7. The method as in claim 1, wherein the device is one of: a data processing device, a portable device, a portable data processing device, a multi touch device, a multi touch portable device, a wireless device, and a cell phone.

8. A machine readable storage medium storing executable program instructions which when executed cause a data processing system to perform a method comprising:

receiving a user input, the user input is one or more input points applied to a touch-sensitive display that is integrated with the data processing system;

creating an event object in response to the user input;

determining whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation;

issuing at least one scroll or gesture call based on invoking the scroll or gesture operation;

responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object; and

responding to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.

9. The medium as in claim 8, further comprising:

rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolled region exceeds a window edge based on the scroll.

10. The medium as in claim 8, further comprising:

attaching scroll indicators to a content edge of the view.

11. The medium as in claim 8, further comprising:

attaching scroll indicators to a window edge of the view.

12. The medium as in claim 8, wherein determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period.

13. The medium as in claim 8, further comprising:

responding to at least one gesture call, if issued, by rotating a view associated with the event object based on receiving a plurality of input points in the form of the user input.

14. The medium as in claim 8, wherein the data processing system is one of: a data processing device, a portable device, a portable data processing device, a multi touch device, a multi touch portable device, a wireless device, and a cell phone.

15. An apparatus, comprising:

means for receiving, through a hardware device, a user input on a touch-sensitive display of the apparatus, the user input is one or more input points applied to the touch-sensitive display that is integrated with the apparatus;

means for creating an event object in response to the user input;

means for determining whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation;

means for issuing at least one scroll or gesture call based on invoking the scroll or gesture operation;

means for responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object; and

means for responding to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.

16. The apparatus as in claim **15**, further comprising:

means for rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll.

17. The apparatus as in claim **15**, further comprising:

means for attaching scroll indicators to a content edge of the window.

18. The apparatus as in claim **15**, further comprising:

means for attaching scroll indicators to the window edge.

19. The apparatus as in claim **15**, wherein determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period.

20. The apparatus as in claim **15**, further comprising:

means for responding to at least one gesture call, if issued, by rotating a view associated with the event object based on receiving a plurality of input points in the form of the user input.

21. The apparatus as in claim **15**, wherein the apparatus is one of: a data processing device, a portable device, a portable data processing device, a multi touch device, a multi touch portable device, a wireless device, and a cell phone.

* * * * *

VIII. CLAIMS APPENDIX

Claims 1-21 are involved in this appeal.

Claim 1 (original) A machine implemented method for scrolling on a touch-sensitive display of a device comprising:

receiving a user input, the user input is one or more input points applied to the touch-sensitive display that is integrated with the device;

creating an event object in response to the user input;

determining whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation;

issuing at least one scroll or gesture call based on invoking the scroll or gesture operation;

responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object based on an amount of a scroll with the scroll stopped at a predetermined position in relation to the user input; and

responding to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.

Claim 2 (original) The method as in claim 1, further comprising:

rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll.

Claim 3 (original) The method as in claim 1, further comprising:

attaching scroll indicators to a content edge of the window.

Claim 4 (original) The method as in claim 1, further comprising:

attaching scroll indicators to the window edge.

Claim 5 (original) The method as in claim 1, wherein determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period.

Claim 6 (original) The method as in claim 1, further comprising:

responding to at least one gesture call, if issued, by rotating a view associated with the event object based on receiving a plurality of input points in the form of the user input.

Claim 7 (original) The method as in claim 1, wherein the device is one of: a data processing device, a portable device, a portable data processing device, a multi touch device, a multi touch portable device, a wireless device, and a cell phone.

Claim 8 (original) A machine readable storage medium storing executable program instructions which when executed cause a data processing system to perform a method comprising:

receiving a user input, the user input is one or more input points applied to a touch-sensitive display that is integrated with the data processing system;

creating an event object in response to the user input;

determining whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation;

issuing at least one scroll or gesture call based on invoking the scroll or gesture operation;

responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object; and

responding to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.

Claim 9 (original) The medium as in claim 8, further comprising:

rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolled region exceeds a window edge based on the scroll.

Claim 10 (original) The medium as in claim 8, further comprising:

attaching scroll indicators to a content edge of the view.

Claim 11 (original) The medium as in claim 8, further comprising:

attaching scroll indicators to a window edge of the view.

Claim 12 (original) The medium as in claim 8, wherein determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period.

Claim 13 (original) The medium as in claim 8, further comprising:

responding to at least one gesture call, if issued, by rotating a view associated with the event object based on receiving a plurality of input points in the form of the user input.

Claim 14 (original) The medium as in claim 8, wherein the data processing system is one of: a data processing device, a portable device, a portable data processing device, a multi touch device, a multi touch portable device, a wireless device, and a cell phone.

Claim 15 (original) An apparatus, comprising:

means for receiving, through a hardware device, a user input on a touch-sensitive display of the apparatus, the user input is one or more input points applied to the touch-sensitive display that is integrated with the apparatus;

means for creating an event object in response to the user input;

means for determining whether the event object invokes a scroll or gesture operation by distinguishing between a single input point applied to the touch-sensitive display that is interpreted as the scroll operation and two or more input points applied to the touch-sensitive display that are interpreted as the gesture operation;

means for issuing at least one scroll or gesture call based on invoking the scroll or gesture operation;

means for responding to at least one scroll call, if issued, by scrolling a window having a view associated with the event object; and

means for responding to at least one gesture call, if issued, by scaling the view associated with the event object based on receiving the two or more input points in the form of the user input.

Claim 16 (original) The apparatus as in claim 15, further comprising:

means for rubberbanding a scrolling region displayed within the window by a predetermined maximum displacement when the scrolling region exceeds a window edge based on the scroll.

Claim 17 (original) The apparatus as in claim 15, further comprising:

means for attaching scroll indicators to a content edge of the window.

Claim 18 (original) The apparatus as in claim 15, further comprising:

means for attaching scroll indicators to the window edge.

Claim 19 (original) The apparatus as in claim 15, wherein determining whether the event object invokes a scroll or gesture operation is based on receiving a drag user input for a certain time period.

Claim 20 (original) The apparatus as in claim 15, further comprising:

means for responding to at least one gesture call, if issued, by rotating a view associated with the event object based on receiving a plurality of input points in the form of the user input.

Claim 21 (original) The apparatus as in claim 15, wherein the apparatus is one of: a data processing device, a portable device, a portable data processing device, a multi touch device, a multi touch portable device, a wireless device, and a cell phone.

CERTIFICATE OF SERVICE

I hereby certify that, on this 3rd day of May, 2016 I filed the foregoing Brief for Appellant Apple Inc. with the Clerk of the United States Court of Appeals for the Federal Circuit via the CM/ECF system, which will send notice of such filing to all registered CM/ECF users.

/s/ William F. Lee
WILLIAM F. LEE
WILMER CUTLER PICKERING
HALE AND DORR LLP
60 State Street
Boston, MA 02109
(617) 526-6000

CERTIFICATE OF COMPLIANCE

Pursuant to Fed. R. App. P. 32(a)(7)(C), the undersigned hereby certifies that this brief complies with the type-volume limitation of Fed. R. App. P. 32(a)(7)(B) and Circuit Rule 32(b).

1. Exclusive of the exempted portions of the brief, as provided in Fed. R. App. P. 32(a)(7)(B), the brief contains 11,650 words.

2. The brief has been prepared in proportionally spaced typeface using Microsoft Word 2010 in 14 point Times New Roman font. As permitted by Fed. R. App. P. 32(a)(7)(C), the undersigned has relied upon the word count feature of this word processing system in preparing this certificate.

/s/ William F. Lee
WILLIAM F. LEE
WILMER CUTLER PICKERING
HALE AND DORR LLP
60 State Street
Boston, MA 02109
(617) 526-6000

May 3, 2016